STA 414/2104:
Probabilistic Machine Learning

Week 13 : Final Exam Review

Murat A. Erdogdu

University of Toronto

Final exam logistic

- Final exam will be held in person on April 29, at 9am-12pm Toronto local time in room MY 150 (all sections).
- Exam will be 100 points in total and 180 mins long. Students are required to be at the exam location at least 10 mins early, with valid identification. Exam will be administered by FAS.
- You can use two optional handwritten A4 aid sheets - double-sided.
- Exam covers all lectures (weeks 1-12), it is closed book/internet.
- You are not responsible for the concepts introduced **only** in suggested readings. However, practicing those would give you a significant advantage.
- A representative practice exam will be posted on the webpage this week.

Probabilistic ML Terminology

The final exam will be on the entire course; however, it will be more weighted towards post-midterm material. For pre-midterm material, refer to the midterm review slides on the website.

- Exponential families
- Directed Graphical Models
- Markov Random Fields
- Message passing
- Belief propagation
- Variable eliminitaion
- Sampling methods
- Markov chain Monte Carlo
- Variational Inference
- EM algorithm
- Probabilistic PCA
- Bayesian regression
- Variational Autoencoders
- Kernel methods
- Gaussian processes
- Diffusion models

KL divergence

We measure the difference between q and p using the **Kullback-Leibler divergence**

$$KL(q(z)||p(z)) = \int q(z) \log \frac{q(z)}{p(z)} dz$$

$$\text{or} = \sum_z q(z) \log \frac{q(z)}{p(z)}$$

Properties of the KL Divergence

- $KL(q||p) \geq 0$
- $KL(q||p) = 0 \Leftrightarrow q = p$
- $KL(q||p) \neq KL(p||q)$
- KL divergence is not a metric, since it's not symmetric

I & M Projection

- I-projection: $q^* = \arg\min_{q \in Q} KL(q||p) = \mathbb{E}_{x \sim q(x)} \log \frac{q(x)}{p(x)}$:
  - $p \approx q \implies KL(q||p)$ small
  - I-projection underestimates support, and does not yield the correct moments.
  - $KL(q||p)$ penalizes q having mass where p has none.
- M-projection: $q^* = \arg\min_{q \in Q} KL(p||q) = \mathbb{E}_{x \sim p(x)} \log \frac{p(x)}{q(x)}$:
  - $p \approx q \implies KL(p||q)$ small
  - $KL(p||q)$ penalizes q missing mass where p has some.
  - M-projection yields a distribution $q(x)$ with the correct mean and covariance.
- One way to proceed is the mean-field approach where we assume:

$$q(x) = \prod_{i \in V} q_i(x_i)$$

the set Q is composed of those distributions that factor out.

Evidence Lower Bound

ELBO is a lower bound on the (log) evidence. Maximizing the ELBO is the same as minimizing $KL(q_\phi(z)||p(z|x))$.

$$
\begin{aligned}
KL(q_\phi(z)||p(z|x)) &= \mathop{\mathbb{E}}_{z\sim q_\phi} \log \frac{q_\phi(z)}{p(z|x)} \\
&= \mathop{\mathbb{E}}_{z\sim q_\phi}\left[\log \left(q_\phi(z) \cdot \frac{p(x)}{p(z,x)} \right) \right] \\
&= \mathop{\mathbb{E}}_{z\sim q_\phi}\left[\log \frac{q_\phi(z)}{p(z,x)} \right] + \mathop{\mathbb{E}}_{z\sim q_\phi} \log p(x) \\
&:= -\mathcal{L}(\phi) + \log p(x)
\end{aligned}
$$

Where $\mathcal{L}(\phi)$ is the **ELBO**:

$$
\mathcal{L}(\phi) = \mathop{\mathbb{E}}_{z\sim q_\phi}\left[\log p(z,x) - \log q_\phi(z) \right]
$$

ELBO

- Rearranging, we get

$$\mathcal{L}(\phi) + KL(q_\phi(z)||p(z|x)) = \log p(x)$$

- Because $KL(q_\phi(z)||p(z|x)) \geq 0$,

$$\mathcal{L}(\phi) \leq \log p(x)$$

- maximizing the ELBO $\Rightarrow$ minimizing $KL(q_\phi(z)||p(z|x))$.

EM Algorithm

- In practice, we are not given a complete dataset $\{\mathbf{X}, \mathbf{Z}\}$, but only incomplete dataset $\{\mathbf{X}\}$.
- Our knowledge about the latent variables is given only by the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \theta)$.
- Because we cannot use the complete data log-likelihood, we can consider expected complete-data log-likelihood:

$$\mathcal{Q}(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \log p(\mathbf{X}, \mathbf{Z}|\theta)$$

- In the E-step, we use the current parameters θ^{old} to compute the posterior over the latent variables $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$.
- In the M-step, we find the revised parameter estimate θ new by maximizing the expected complete log-likelihood:

$$\theta^{new} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{old})$$

EM Algorithm

- Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\theta)$ over observed and latent variables, the goal is to maximize the likelihood function $p(\mathbf{X}|\theta)$ with respect to θ.
- Initialize parameters θ^{old}
- **E-step:** Compute the posterior $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$ and $\mathcal{Q}(\theta, \theta^{old})$
- **M-step:** Find the new estimate of parameters θ^{new}:

$$\theta^{new} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{old})$$

where

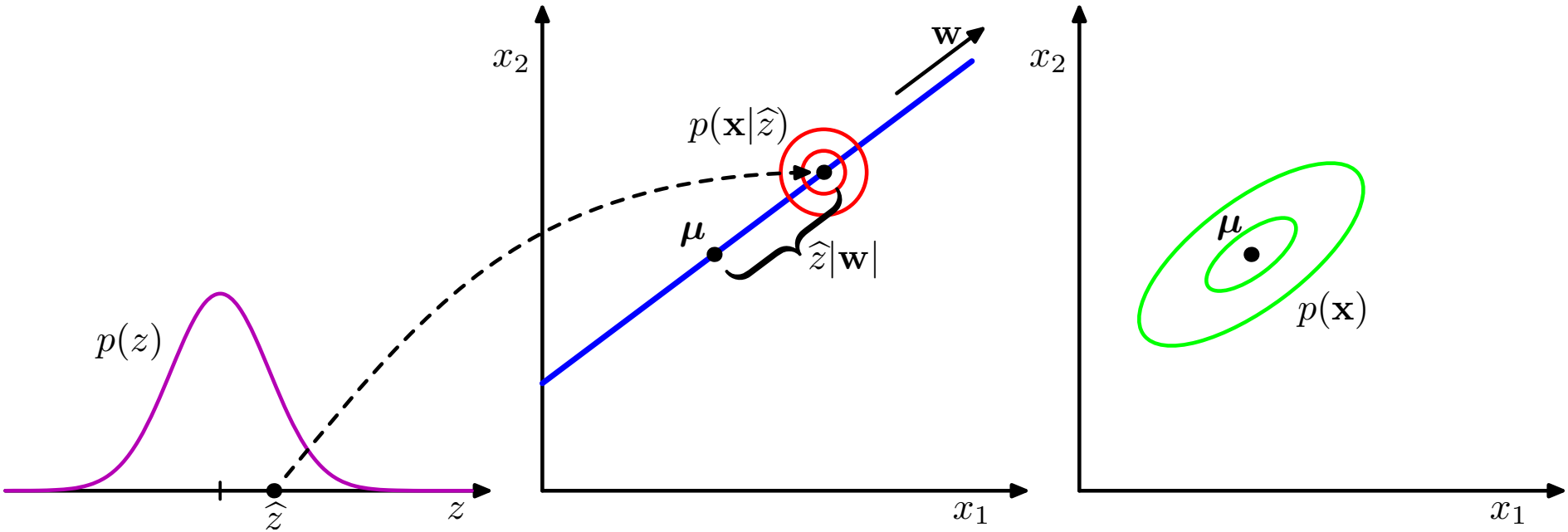
$$\mathcal{Q}(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \log p(\mathbf{X}, \mathbf{Z}|\theta).$$

Probabilistic PCA

- Similar to the Gaussian mixture model, we assume continuous, Gaussian latent variables:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{x} \,|\, \mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$$



Probabilistic PCA - Maximum Likelihood

- $p(\mathbf{x} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2)$ will be Gaussian (confirm this), so we just need

$$\mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \epsilon] = \boldsymbol{\mu}$$

$$\begin{aligned}
\text{Cov}[\mathbf{x}] &= \mathbb{E}[(\mathbf{W}\mathbf{z} + \epsilon)(\mathbf{W}\mathbf{z} + \epsilon)^\top] \\
&= \mathbb{E}[\mathbf{W}\mathbf{z}\mathbf{z}^\top\mathbf{W}^\top] + \text{Cov}[\epsilon] \\
&= \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}
\end{aligned}$$

  Thus, the posterior mean

$$\mathbb{E}[\mathbf{z} \mid \mathbf{x}] = \left(\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{W}^\top(\mathbf{x} - \boldsymbol{\mu})$$

- To perform MLE, we need to maximize the following:

$$\max_{\mathbf{W},\boldsymbol{\mu},\sigma^2} \log p(\mathbf{x} \mid \mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \max_{\mathbf{W},\boldsymbol{\mu},\sigma^2} \log \int p(\mathbf{x} \mid \mathbf{z}, \mathbf{W}, \boldsymbol{\mu}, \sigma^2)p(\mathbf{z})\, d\mathbf{z}$$

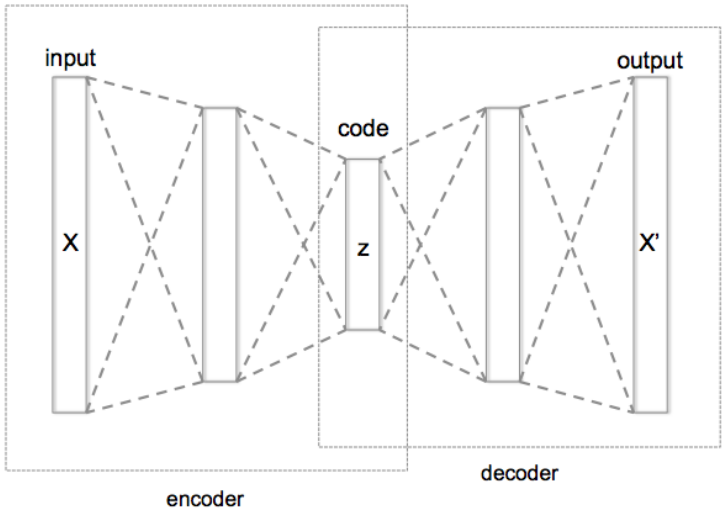
Autoencoders

Autoencoders reconstruct their input via an encoder and a decoder.

- **Encoder**: $g(x) = z \in F, \quad x \in X$
- **Decoder**: $f(z) = \tilde{x} \in X$
- where X is the data space, and F is the feature (latent) space.
- z is the code, compressed representation of the input, x. It is important that this code is a bottleneck, i.e. that

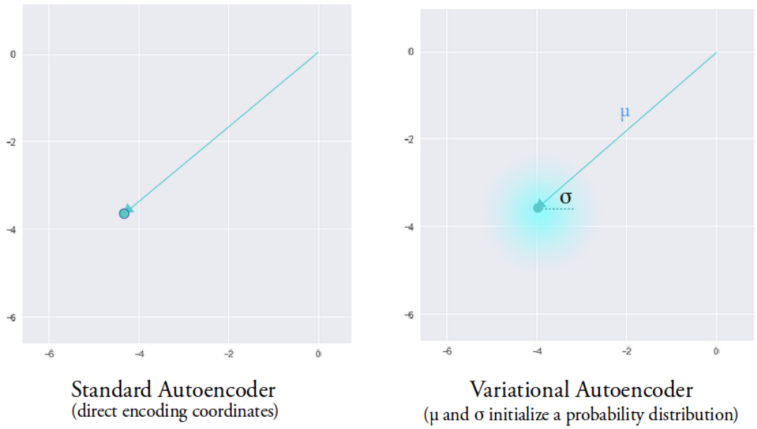
$$\dim F \ll \dim X$$

- Goal: $\tilde{x} = f(g(x)) \approx x$.

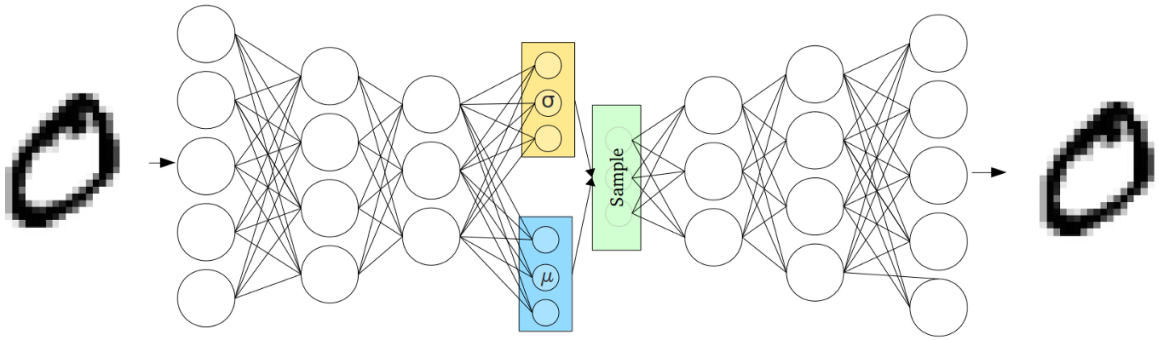


Variational Autoencoders

- The mean μ controls where encoding of input is centered while the standard deviation controls how much can the encoding vary.



- Encodings are generated at random from the "circle", the decoder learns that all nearby points refer to the same input.



VAE vs Amortized VAE Pipeline

- For a given input (or minibatch) x_i,

  - **Standard VAE**
  - Sample $z_i \sim q_{\phi_i}(z|x_i) = \mathcal{N}(\mu_i, \sigma_i^2 I)$.

  - **Amortized VAE**
  - Sample $z_i \sim q_\phi(z|x_i) = \mathcal{N}(\mu_\phi(x_i), \Sigma_\phi(x_i))$

- Run the code through decoder and get likelihood: $p_\theta(x|z)$.
- Compute the loss function (-ELBO):
  $L(x;\theta,\phi) = -E_{z_\phi \sim q_\phi}\Big[\log p_\theta(x|z)\Big] + KL(q_\phi(z|x)||p(z))$
- Use gradient-based optimization to backpropogate $\partial_\theta L$, $\partial_\phi L$

Bayesian Linear Regression

- **Prior distribution:** $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{S})$
- **Model (Likelihood):** $y \,|\, \mathbf{x}, \mathbf{w} \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}),\ \sigma^2)$
- Assuming fixed/known $\mathbf{S}$ and σ^2.

Bayesian Linear Regression: Posterior

- **Deriving the posterior distribution:**

$$\begin{aligned}
\log p(\mathbf{w} \mid \mathcal{D}) &= \log p(\mathbf{w}) + \log p(\mathcal{D} \mid \mathbf{w}) + \text{const} \\
&= -\tfrac{1}{2}\mathbf{w}^\top \mathbf{S}^{-1}\mathbf{w} - \frac{1}{2\sigma^2}\|\boldsymbol{\Psi}\mathbf{w} - \boldsymbol{y}\|^2 + \text{const} \\
&= -\tfrac{1}{2}\mathbf{w}^\top \mathbf{S}^{-1}\mathbf{w} - \frac{1}{2\sigma^2}\left(\mathbf{w}^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi}\mathbf{w} - 2\boldsymbol{y}^\top \boldsymbol{\Psi}\mathbf{w} + \boldsymbol{y}^\top \boldsymbol{y}\right) + \text{const} \\
&= -\tfrac{1}{2}\mathbf{w}^\top \left(\sigma^{-2}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \mathbf{S}^{-1}\right)\mathbf{w} + \frac{1}{\sigma^2}\boldsymbol{y}^\top \boldsymbol{\Psi}\mathbf{w} + \text{const} \text{ (complete the square!)}
\end{aligned}$$

Thus $\mathbf{w} \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top \boldsymbol{y}$$

$$\boldsymbol{\Sigma} = \left(\sigma^{-2}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \mathbf{S}^{-1}\right)^{-1}$$

Gaussian processes

- We have the linear model

$$y \mid \mathbf{x} \sim \mathcal{N}(\hat{y}(\mathbf{x}),\ \sigma^2) \qquad \hat{y}(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x})$$

- Given N independent observations, we have

$$\mathbf{y} \mid \hat{\mathbf{y}} \sim \mathcal{N}(\hat{\mathbf{y}},\ \sigma^2 \mathbf{I}_N), \qquad \hat{\mathbf{y}} \sim \mathcal{N}(0, \boldsymbol{K}).$$

- Therefore the marginal of $\mathbf{y}$ is given by

$$\mathbf{y} \sim \mathcal{N}(0, \boldsymbol{C}) \qquad \boldsymbol{C} = \boldsymbol{K} + \sigma^2 \mathbf{I}_N$$

where the corresponding kernel is

$$c(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \frac{1}{\alpha} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sigma^2 \delta(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$\delta(\mathbf{x}, \mathbf{x}') = 1$ if $\mathbf{x} = \mathbf{x}'$ and $\delta(\mathbf{x}, \mathbf{x}') = 0$ otherwise.

Gaussian processes

- Denote now $\mathbf{y}_N = (y^{(1)}, y^{(2)}, ..., y^{(N)})$.
- We have the marginal of $\mathbf{y}_N$ given by

$$\mathbf{y}_N \sim \mathcal{N}(0, \boldsymbol{C}_N) \qquad \boldsymbol{C}_N = \boldsymbol{K}_N + \sigma^2 \mathbf{I}_N.$$

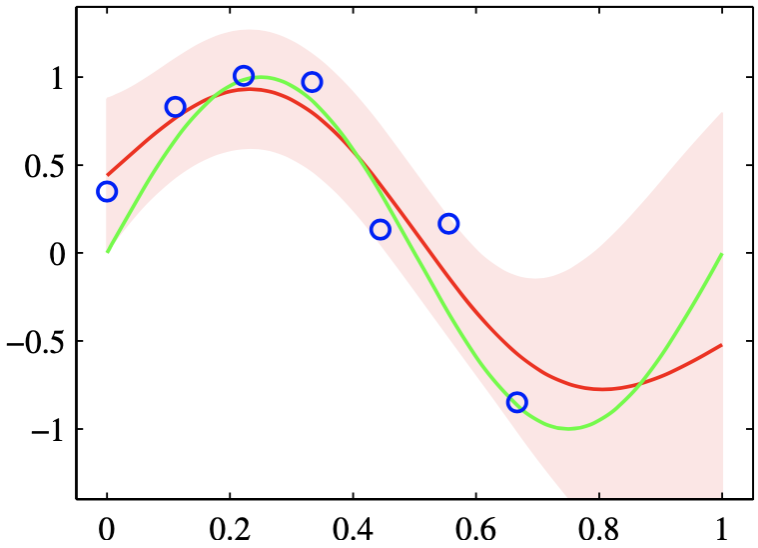
- This reflects the two Gaussian sources of randomness.

Goal: We want to predict for a new output $y^{(N+1)}$ given $\mathbf{x}^{(N+1)}$.

- We showed: Since y_{N+1} is multivariate Gaussian, $y^{(N+1)} \,|\, \mathbf{y}_N$ is also Gaussian with mean and covariance

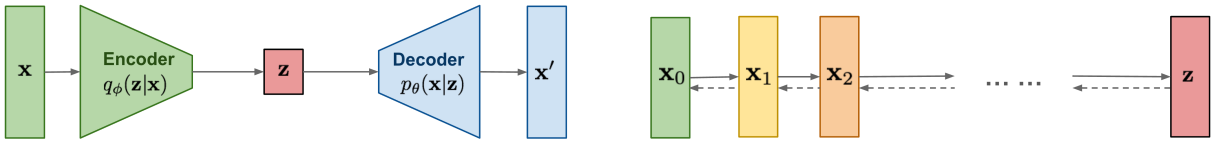
$$m(\mathbf{x}^{(N+1)}) = \mathbf{k}^T \boldsymbol{C}_N^{-1} \mathbf{y}_N \qquad \sigma^2(\mathbf{x}^{(N+1)}) = c - \mathbf{k}^T \boldsymbol{C}_N^{-1} \mathbf{k}$$

GPs for regression



- The green curve: the true sinusoid from which the data points, shown in blue, are obtained with additional of Gaussian noise.
- The red line: mean of the Gaussian process predictive distribution.
- The shaded region: plus and minus two standard deviations.

VAE vs Diffusion Model



- Diffusion models and VAEs both map to isotropic Gaussian.
- The latent space has the same dimension as the input space in DMs. In VAEs, it is smaller dimensional.
- The forward process is the encoder, which is **fixed**. This is trained in VAEs.
- The reverse process is the decoder, which is **trained**, similar to the VAEs.

Closing remarks

Continuing with machine learning:

- Courses
  - CSC 421/2516, "Neural Networks and Deep Learning"
  - CSC 2515, "Machine Learning"
  - CSC 2532, "Statistical Learning Theory"
  - CSC 2541, "Neural Network Training Dynamics"
  - Topics courses (varies from year to year): Reinforcement Learning, Algorithmic Fairness, Computer Vision w/ ML, NLP w/ ML, Health w/ ML etc.
  - Learn Statistics!
- Videos from top ML conferences (NeurIPS, ICML, ICLR)
- Try to reproduce results from papers
  - If they've released code, you can use that as a guide if you get stuck.
- Lots of excellent free resources available online!

Summary

- Review lectures.
- Understand **derivations**.
- Solve the practice final.
- Fill out course evaluations!
- Thanks!