6.390 Midterm Review

Spring 2025

Haley Nakamura

Topic 1: IntroML

- ML: making predictions from data
- Training / validation / testing data
- Supervised learning
- Hypothesis / hypothesis class
- Learning algorithm
- Evaluation metrics
  - Loss functions
  - Overfitting vs. underfitting

$$\mathcal{D}_{\text{train}} = \left\{ \left(x^{(1)}, y^{(1)} \right), \ldots, \left(x^{(n)}, y^{(n)} \right) \right\}$$

$$x \rightarrow \boxed{h} \rightarrow y$$

$$\mathcal{D}_{\text{train}} \longrightarrow \boxed{\text{learning alg } (\mathcal{H})} \longrightarrow h$$

$$\mathcal{E}_{\text{train}}(h; \Theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(x^{(i)}; \Theta), y^{(i)})$$

Topic 2: Linear Regression

- Label: continuous real number
- Linear hypothesis class
- Objective function (e.g. MSE)
- Closed-form/analytical solution (optimal)
- What if $X^T X$ is not invertible?
  - Objective function has “half-pipe” shape instead of “bowl” shape
  - Cases:
    - More features than data points
    - Features are linearly dependent

$$y = h(x; \theta, \theta_0) = \theta^T x + \theta_0$$

$$J(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} + \theta_0 - y^{(i)} \right)^2$$

$$\theta^* = \left(X^T X \right)^{-1} X^T Y$$

Topic 2: Linear Regression

- Add regularizer term
  - Generalizable
- λ as a hyperparameter
- Cross-validation

$$J_{\text{ridge}}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} + \theta_0 - y^{(i)} \right)^2 + \lambda \|\theta\|^2$$

$$\theta_{\text{ridge}} = \left(X^T X + n\lambda I \right)^{-1} X^T Y$$

Topic 3: Gradient Descent

- Gradient vector (analytically and conceptually)
- Algorithm and update step $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta J(\theta)|_{\theta=\theta^{(t)}}$
- η as a hyperparameter
- Termination criterion
- Smooth, convex, sufficiently small learning rate, enough iterations, global min exists?
  - Converge to global minimum!
- What if these assumptions are violated?
- Stochastic gradient descent (what changes?)

Topic 4: Classification

$$z = \theta^T x + \theta_0$$

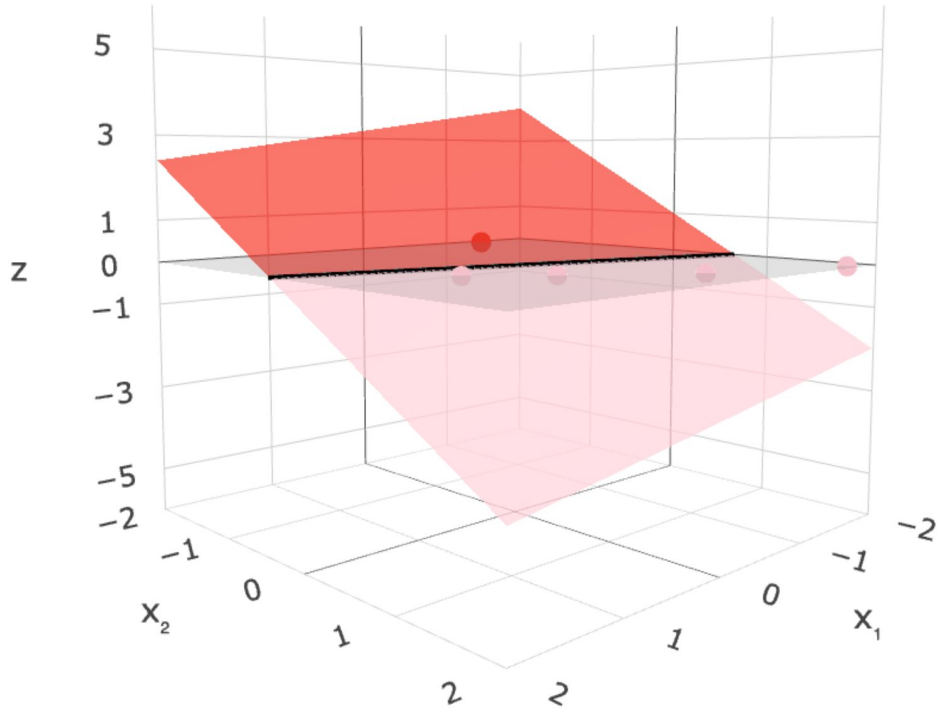
$$h(x; \theta, \theta_0) = \text{step}(\theta^T x + \theta_0)$$

- Binary classification: {1, 0}
- Linear classifier
  - Normal vector
- Logistic classifier
  - Sigmoid, NLL Loss
  - Magnitude of parameters?
- Multiclass:
  - Pick one: softmax, NLLM Loss
  - Pick many: multiple sigmoids

Topic 4: Classification

$$h(x;\theta,\theta_0) = \text{step}(\theta^T x + \theta_0)$$

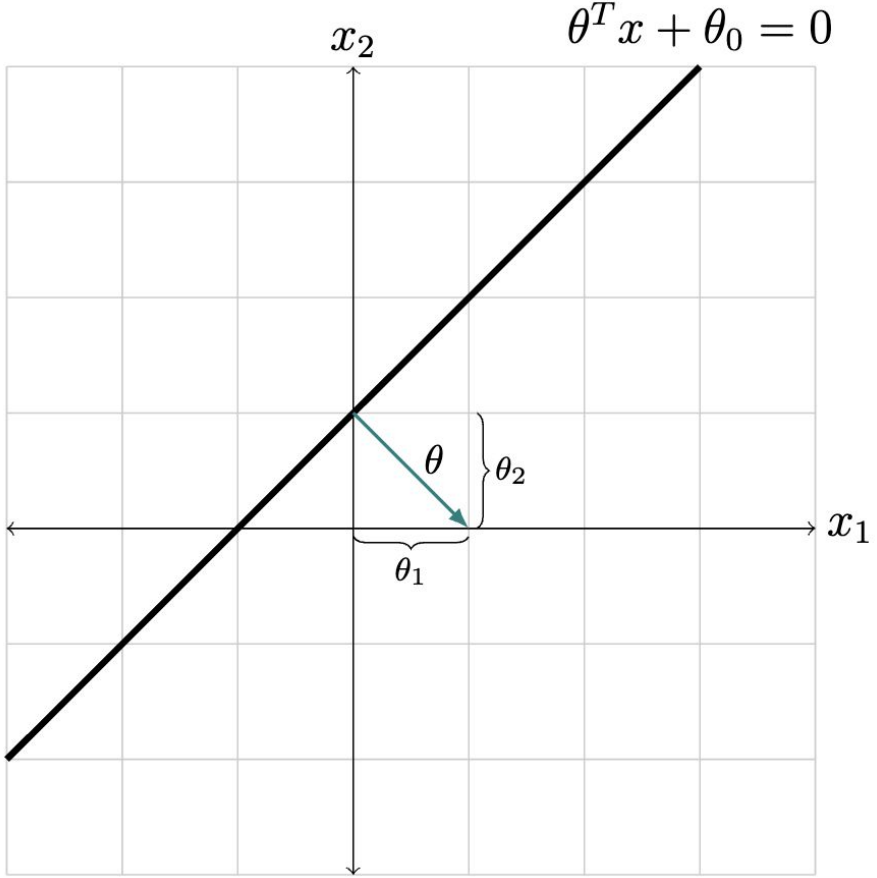
- Binary classification: {1, 0}
- Linear classifier
  - Normal vector
- Logistic classifier
  - Sigmoid, NLL Loss
  - Magnitude of parameters?
- Multiclass:
  - Pick one: softmax, NLLM Loss
  - Pick many: multiple sigmoids



Topic 4: Classification

$$h(x; \theta, \theta_0) = \text{step}(\theta^T x + \theta_0)$$

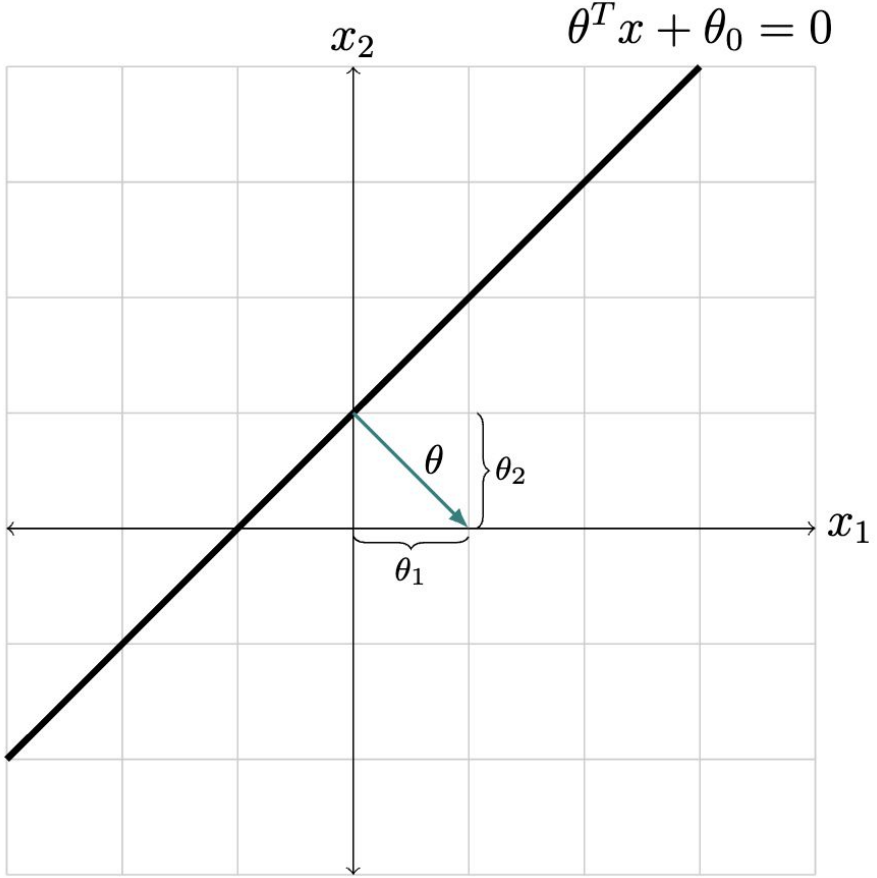
- Binary classification: {1, 0}
- Linear classifier
  - Normal vector
- Logistic classifier
  - Sigmoid, NLL Loss
  - Magnitude of parameters?
- Multiclass:
  - Pick one: softmax, NLLM Loss
  - Pick many: multiple sigmoids



Topic 4: Classification

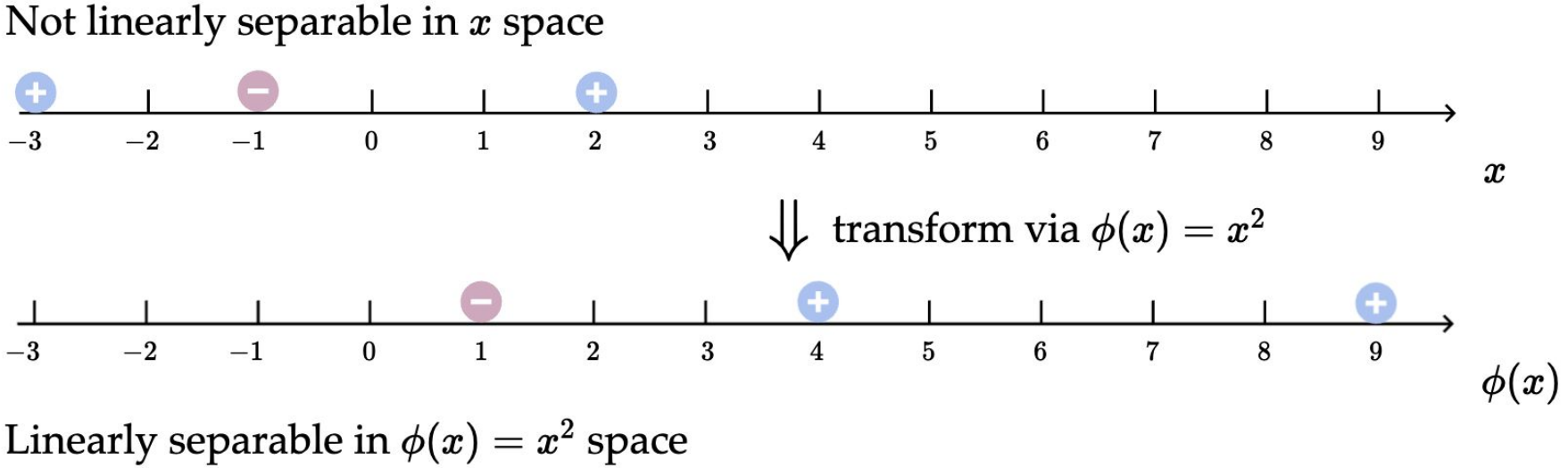
$$h(x;\theta,\theta_0) = \sigma(\theta^T x + \theta_0)$$

- Binary classification: {1, 0}
- Linear classifier
  - Normal vector
- Logistic classifier
  - Sigmoid, NLL Loss
  - Magnitude of parameters?
- Multiclass:
  - Pick one: softmax, NLLM Loss
  - Pick many: multiple sigmoids



Topic 5: Features

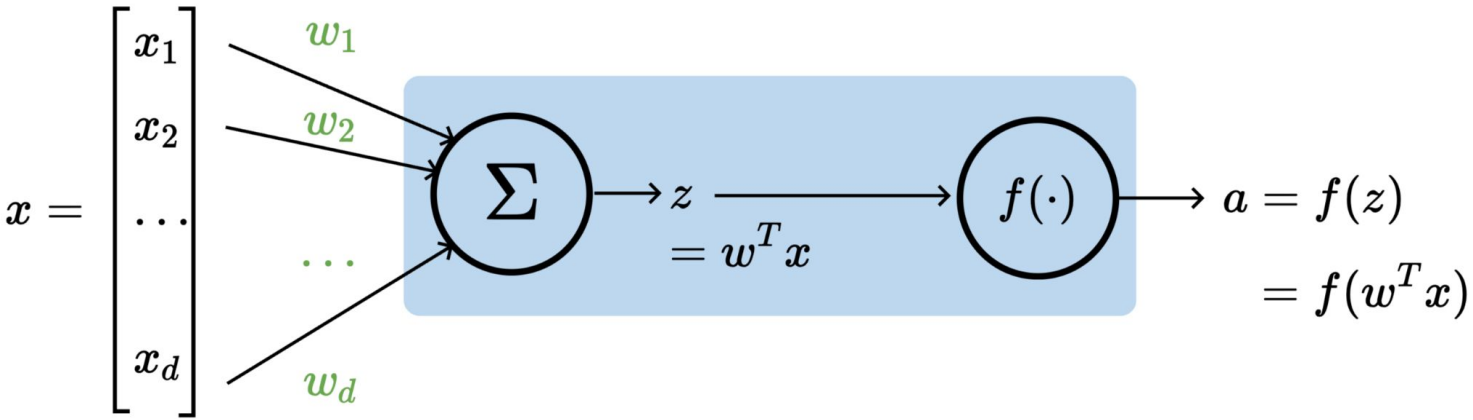
- Nonlinear feature transformations
- (k^{th} order) Polynomial basis



Topic 6: Neural Networks

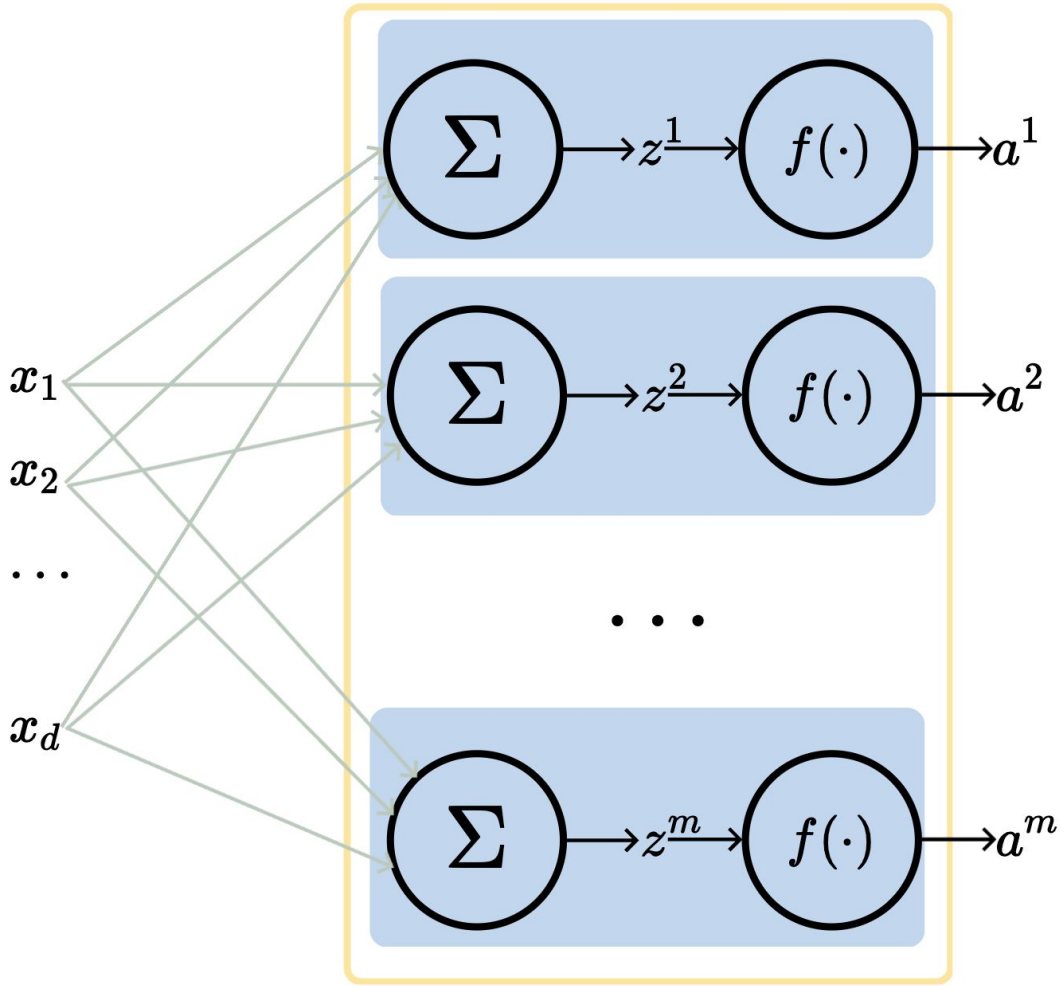
- Computation graph
  - Neurons → Layers

    → Network



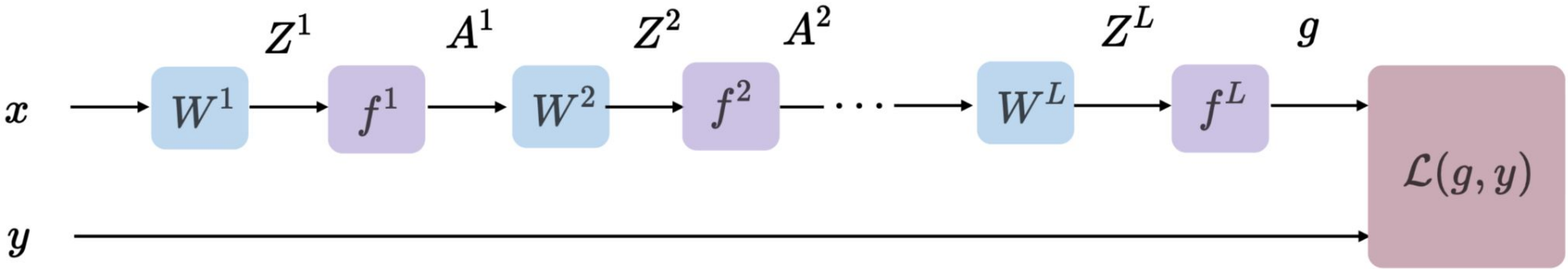
Topic 6: Neural Networks

- Computation graph
  - Neurons → Layers
    → Network



Topic 6: Neural Networks

- Computation graph
  - Neurons → Layers
    → Network
- Network as a hypothesis
  (forward-pass)

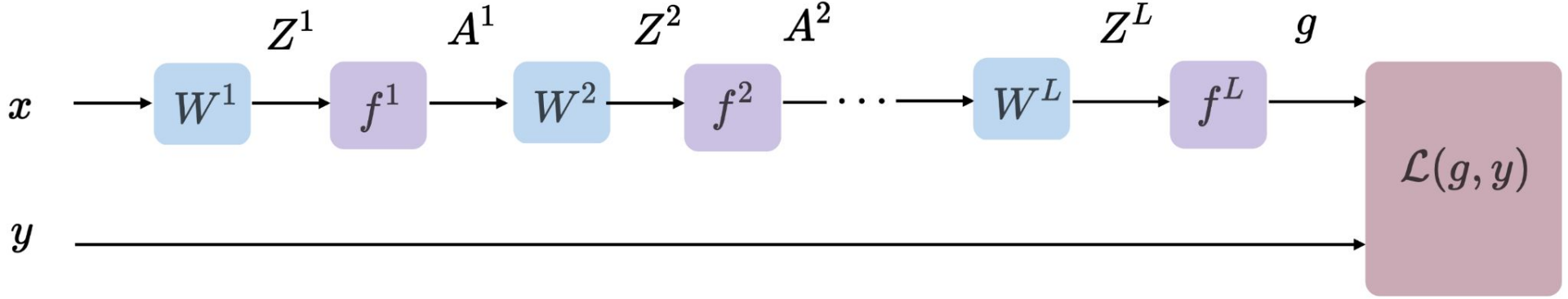


Topic 6: Neural Networks

- Output layer design
  - Dimension (# output neurons), activation function, loss function
- Hand-crafting weights to match a function

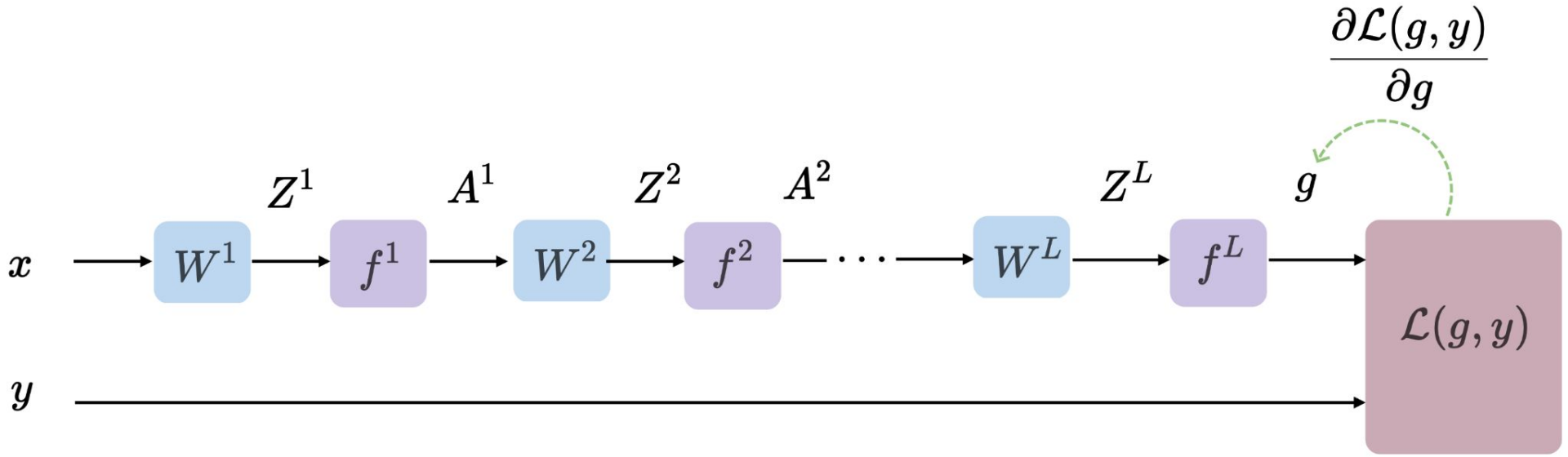
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



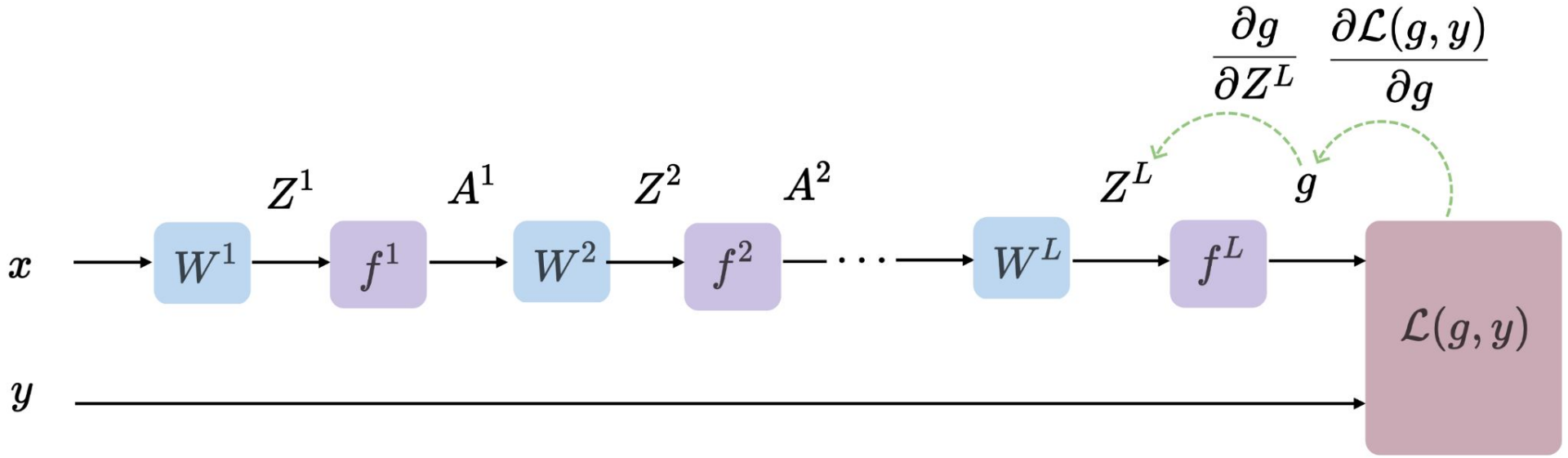
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



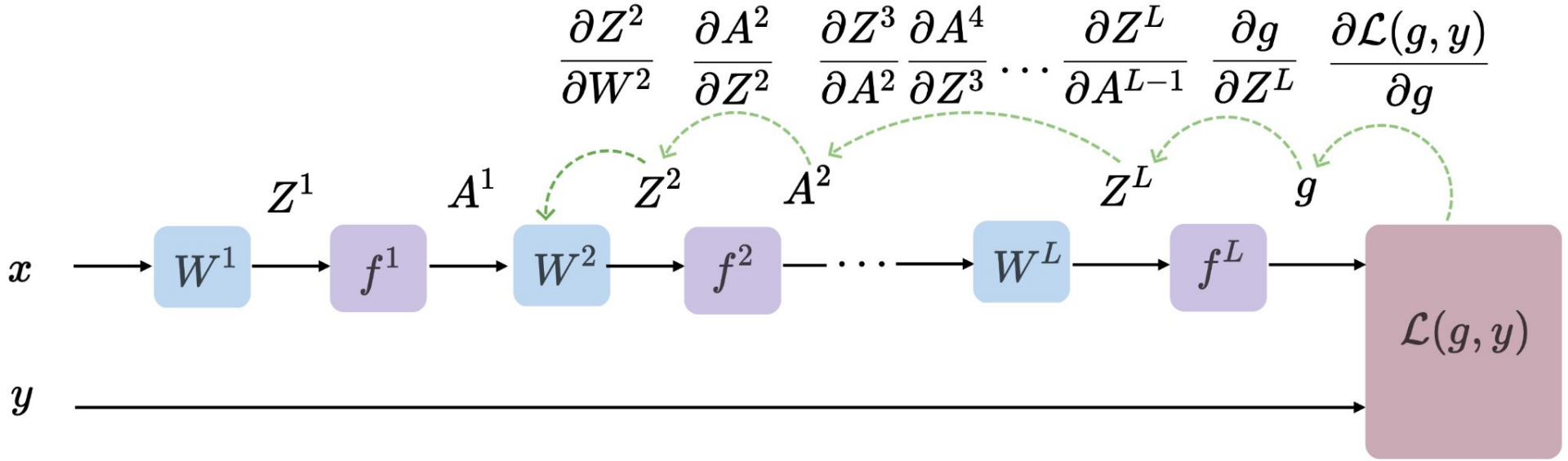
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



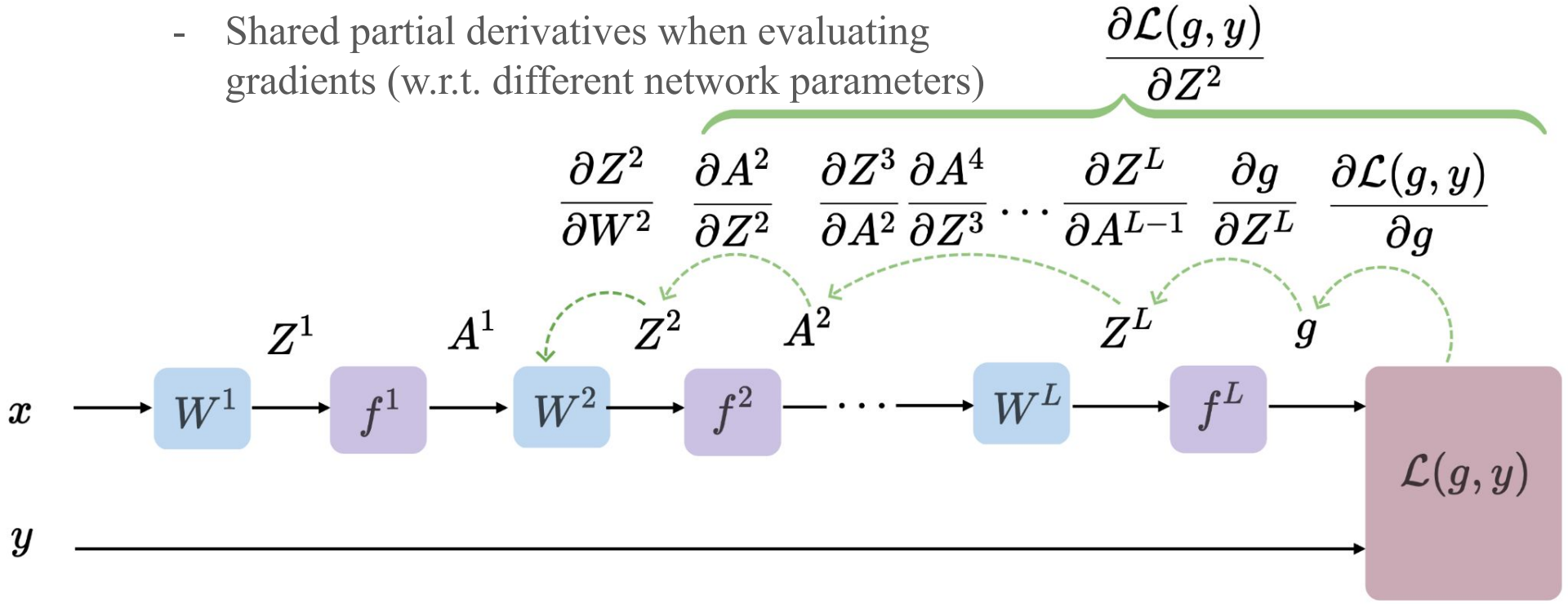
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



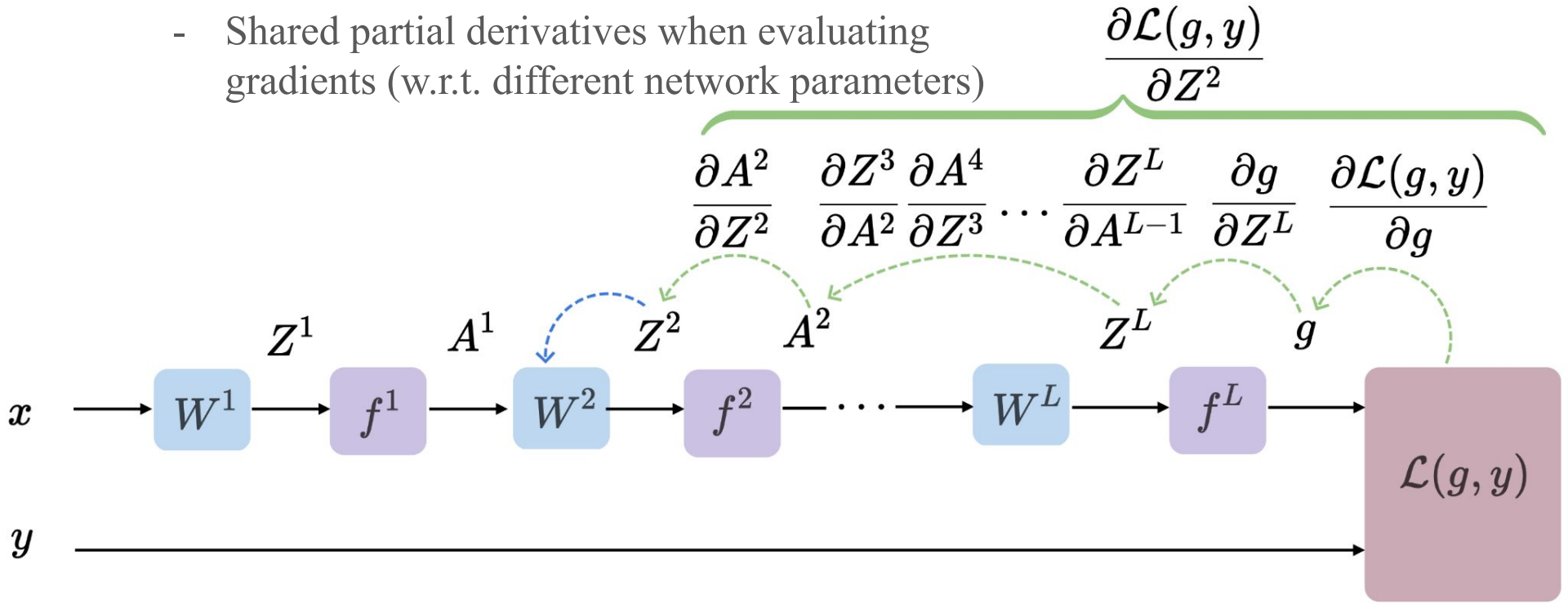
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



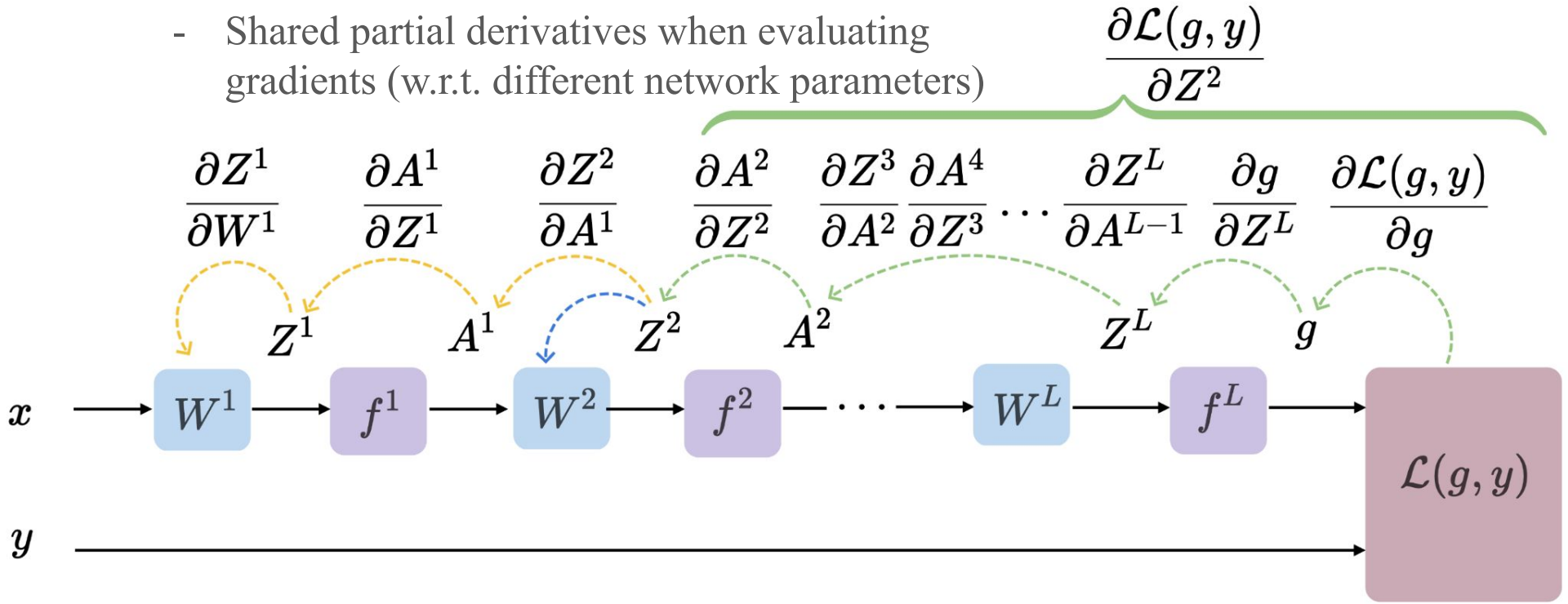
Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



Topic 6: Neural Networks

- Backward-pass (backpropagation)
  - Shared partial derivatives when evaluating gradients (w.r.t. different network parameters)



Q&A