Optimization of deep networks

Data

- Input: $\{\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}\}$
- Label: $\{\mathbf{y}_0, \ldots, \mathbf{y}_{N-1}\}$
- Dataset: $D = \{(\mathbf{x}_0, \mathbf{y}_0), \ldots, (\mathbf{x}_{N-1}, \mathbf{y}_{N-1})\}$

( , dog)

( , dog)

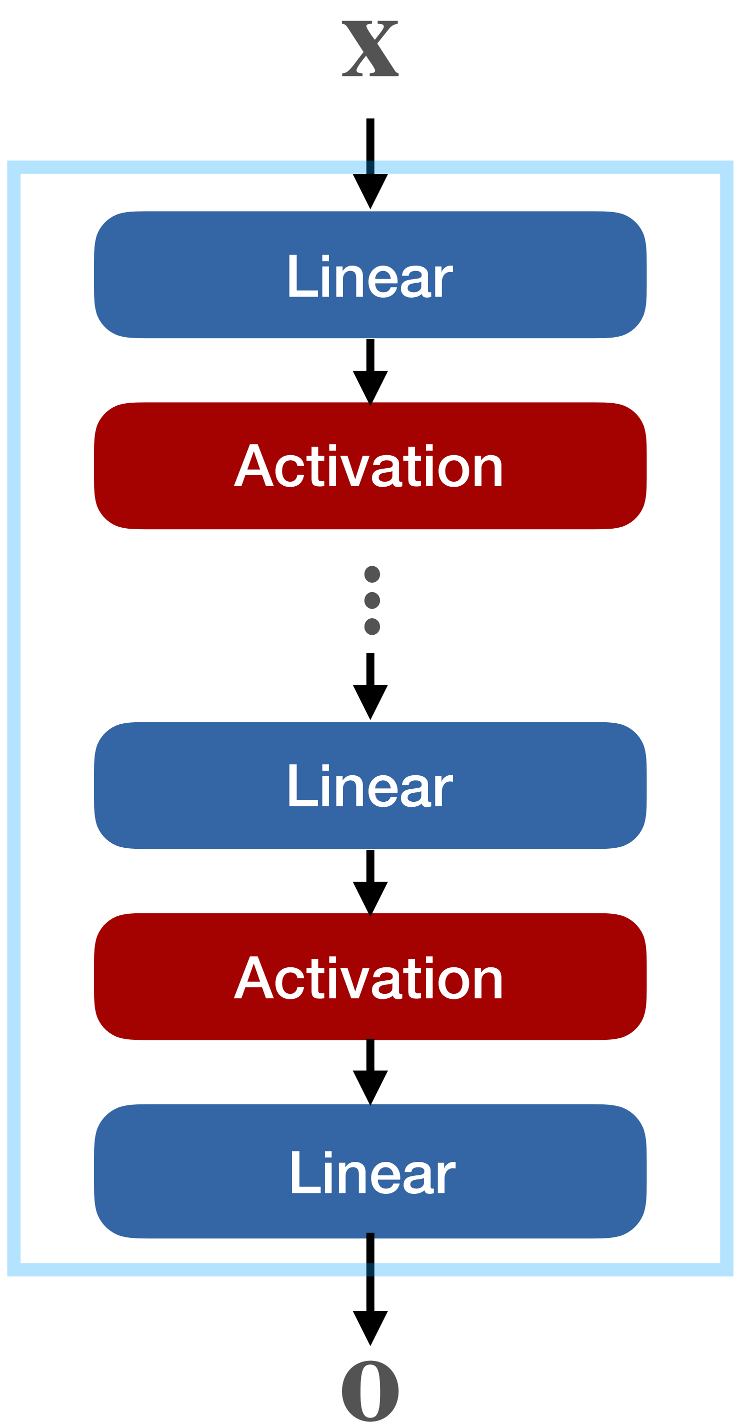
( , cat)

( , cat)

⋮

Model

- Deep network $f : (\mathbf{x}, \theta) \rightarrow \mathbf{o}$
  - Layers of computation
  - Parameters θ
  - Differentiable computation graph

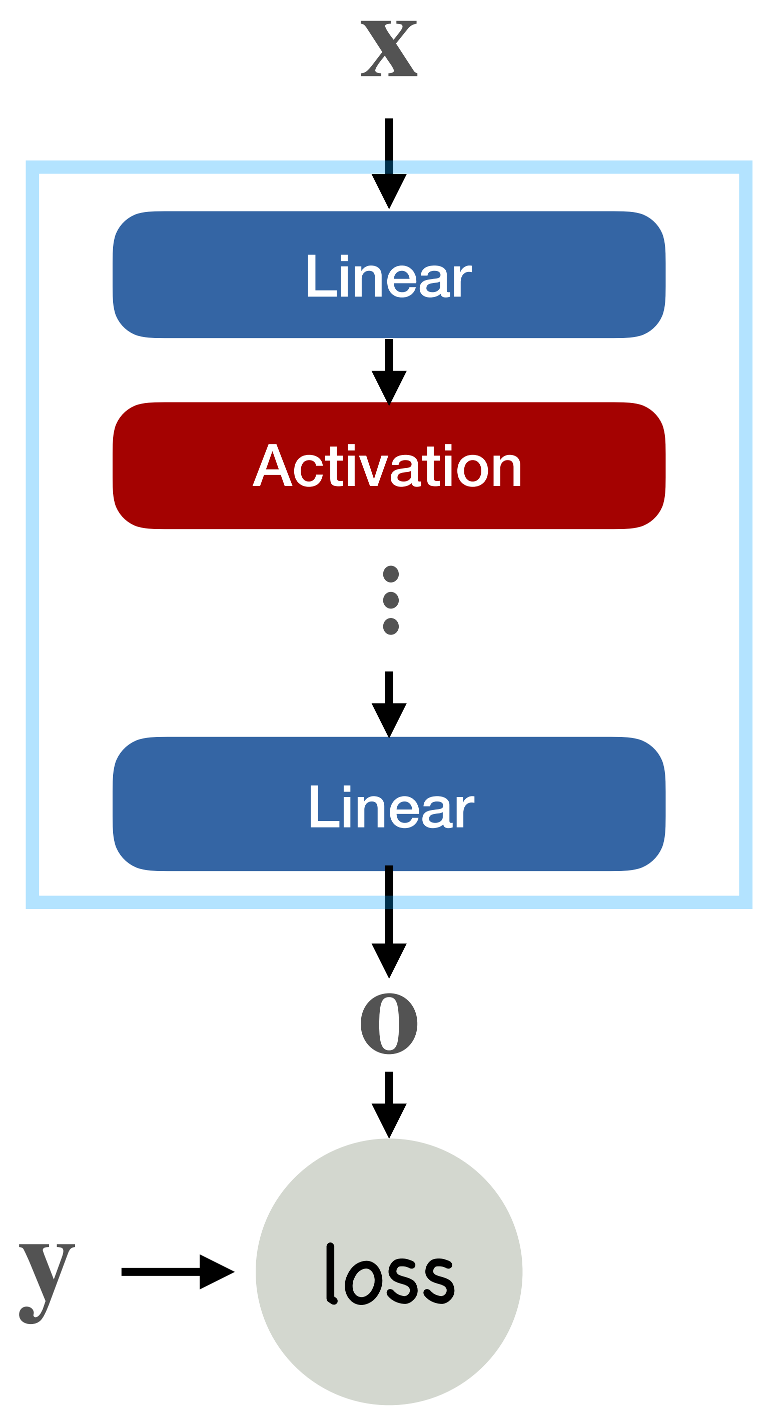


Loss

- Differentiable $\ell(\mathbf{o}, \mathbf{y})$
- Regression
    - Distance norm

      $$\ell(\mathbf{o}, \mathbf{y}) = \|\mathbf{o} - \mathbf{y}\|$$
- Classification
    - Cross Entropy

      $$\ell(\mathbf{o}, y) = -\log p(y)$$
- Over training dataset
    - $L(\theta) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim D}[\ell(f(\mathbf{x}, \theta), \mathbf{y})]$

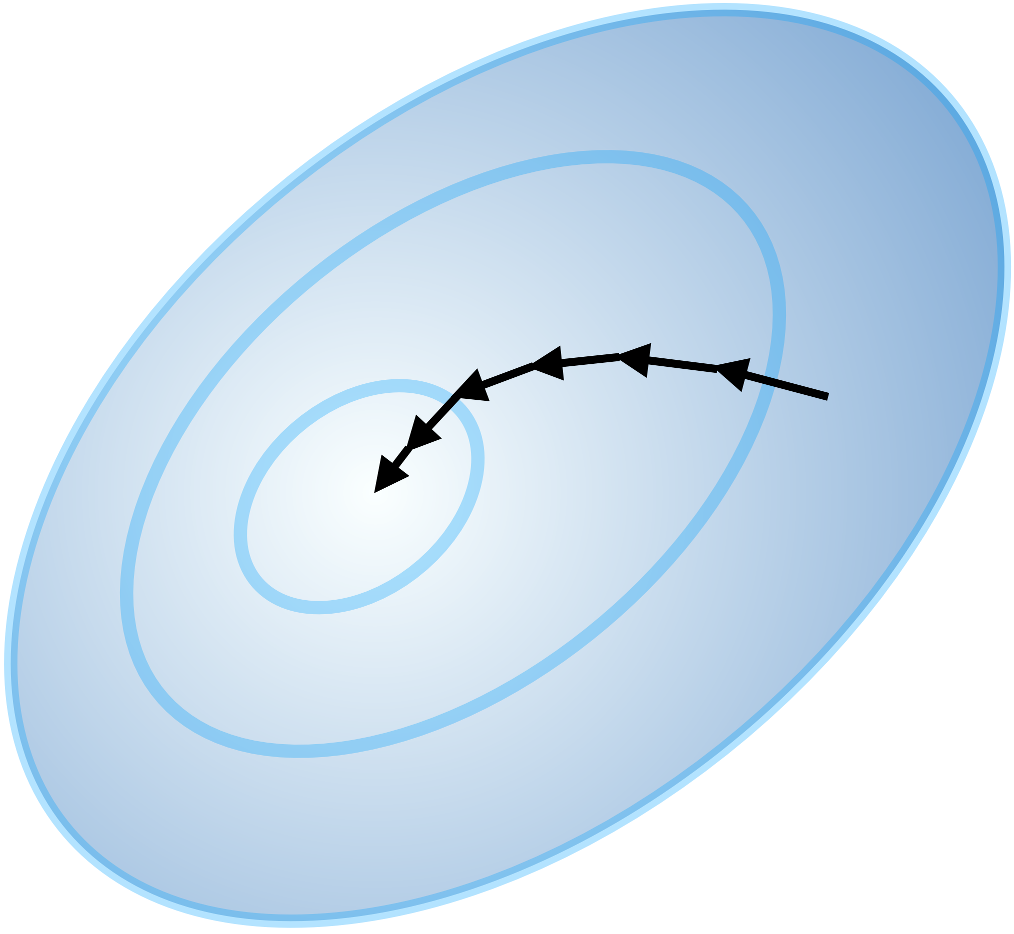


Optimization

- Minimize $L(\theta)$

Gradient Descent

- Repeat until convergence:
  - $\theta := \theta - \epsilon \frac{dL(\theta)}{d\theta}$



Issue with Gradient Descent

- Slow to compute gradient
  - $\frac{dL(\theta)}{d\theta} = \mathbb{E}_{\mathbf{x},\mathbf{y} \in D}[\frac{d\ell(f(\mathbf{x}, \theta), \mathbf{y})}{d\theta}]$