Advanced Machine Learning

Likhit Nayak

What is optimization in deep learning?

The goal of traditional optimization algorithms is to minimize the cost function ***J.***

In deep learning, we care more about some performance measure ***P***, which is measured on some test data. Indirectly, we are trying to optimize ***P*** and reduce the cost function ***J**** in the hope that it will reduce ***P***.

$$J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\mathrm{y})\sim p_{\mathrm{data}}} L(f(\boldsymbol{x};\boldsymbol{\theta}), y)$$

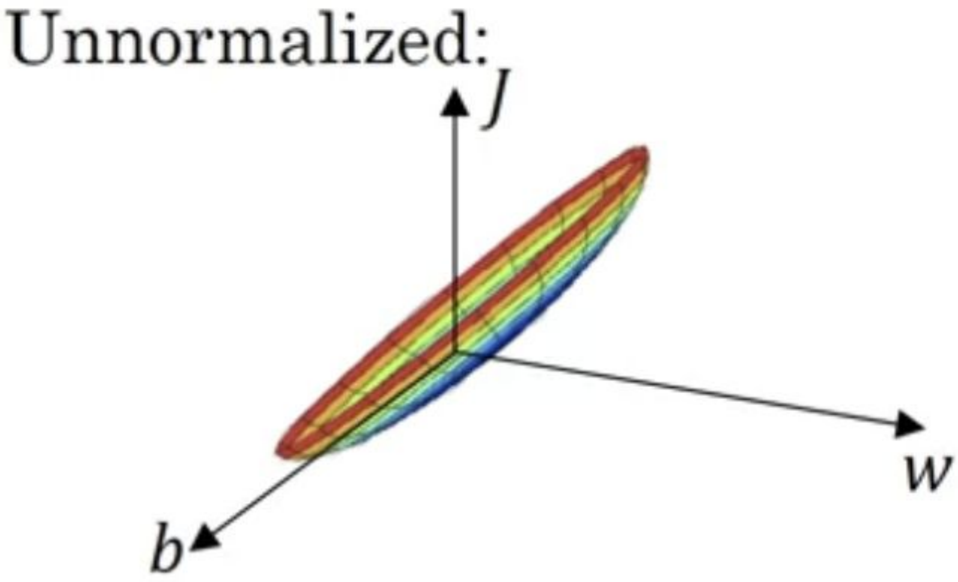
Empirical risk minimization

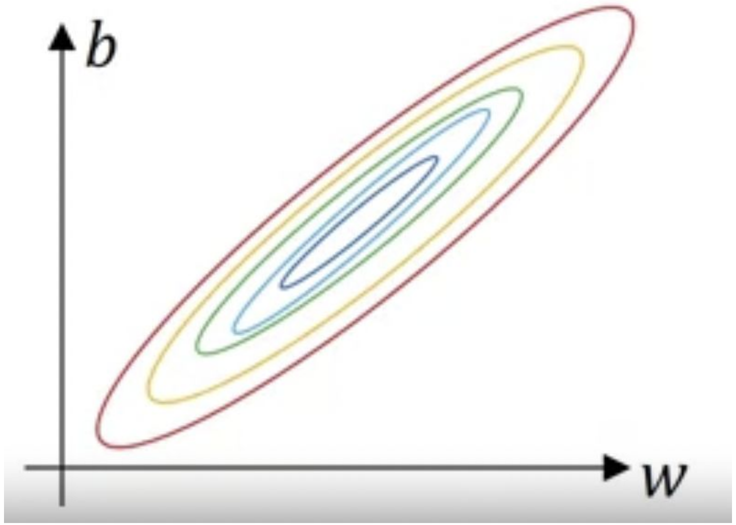
When we don't know $\boldsymbol{p}_{data}$, we try to minimize the loss over a set of training examples:

$$\mathbb{E}_{\boldsymbol{x},\mathrm{y}\sim\hat{p}_{\mathrm{data}}(\boldsymbol{x},y)}[L(f(\boldsymbol{x};\boldsymbol{\theta}),y)] = \frac{1}{m}\sum_{i=1}^{m} L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}),y^{(i)})$$

This is called **empirical risk minimization -** works by minimizing the average training error

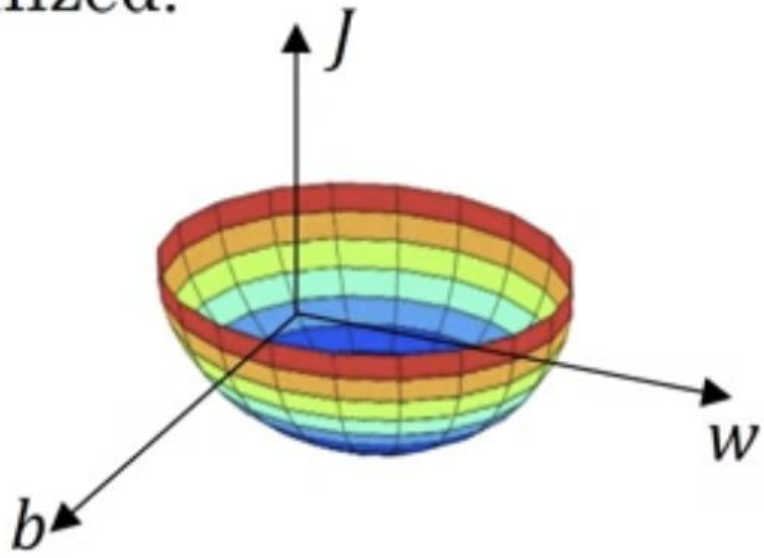
Normalizing inputs





Normalizing inputs

Normalized:



Zero-one loss function

One of the simplest loss functions which literally counts the number of mistakes made by the learned function ***h:***

$$\mathcal{L}_{0/1}(h) = \frac{1}{n}\sum_{i=1}^{n} \delta_{h(\mathbf{x}_i)\neq y_i}, \text{ where } \delta_{h(\mathbf{x}_i)\neq y_i} = \begin{cases} 1, & \text{if } h(\mathbf{x}_i) \neq y_i \\ 0, & \text{o.w.} \end{cases}$$

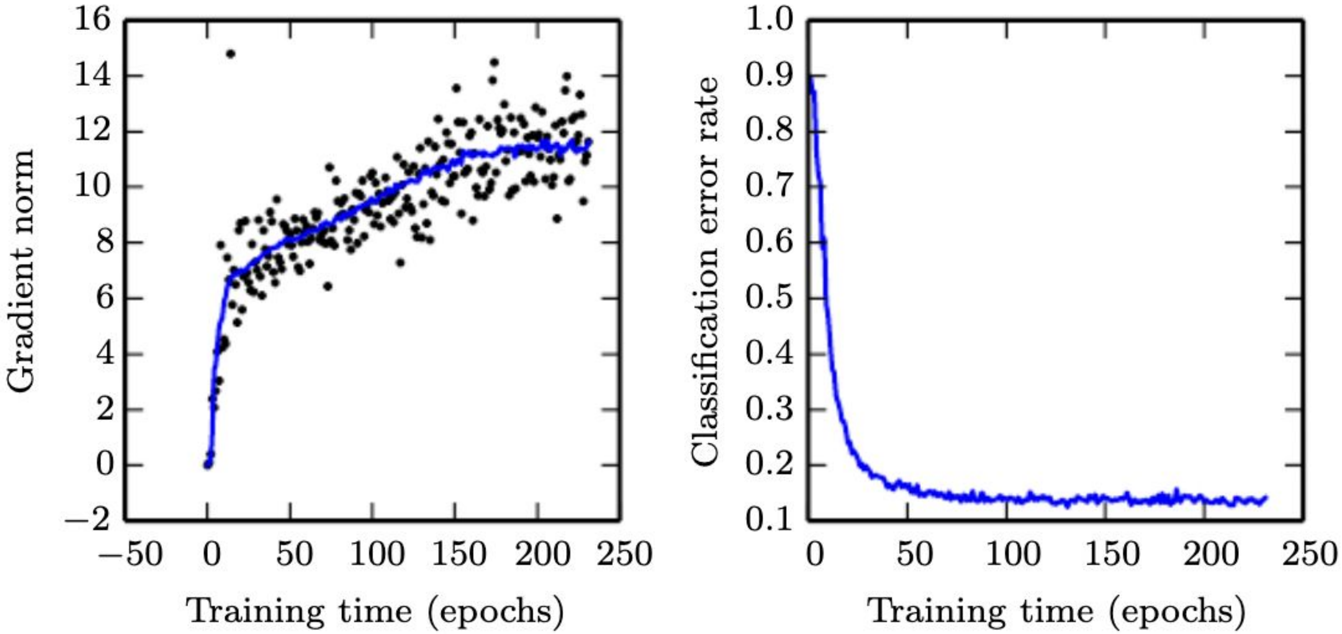
Minibatch Optimization

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathrm{y}\sim\hat{p}_{\mathrm{data}}} \log p_{\mathrm{model}}(\boldsymbol{x}, y; \boldsymbol{\theta}).$$

1. Less accurate than full-batch optimization
2. Faster than full-batch optimization
3. Efficient utilizes multicore architectures
4. Regularizing effect

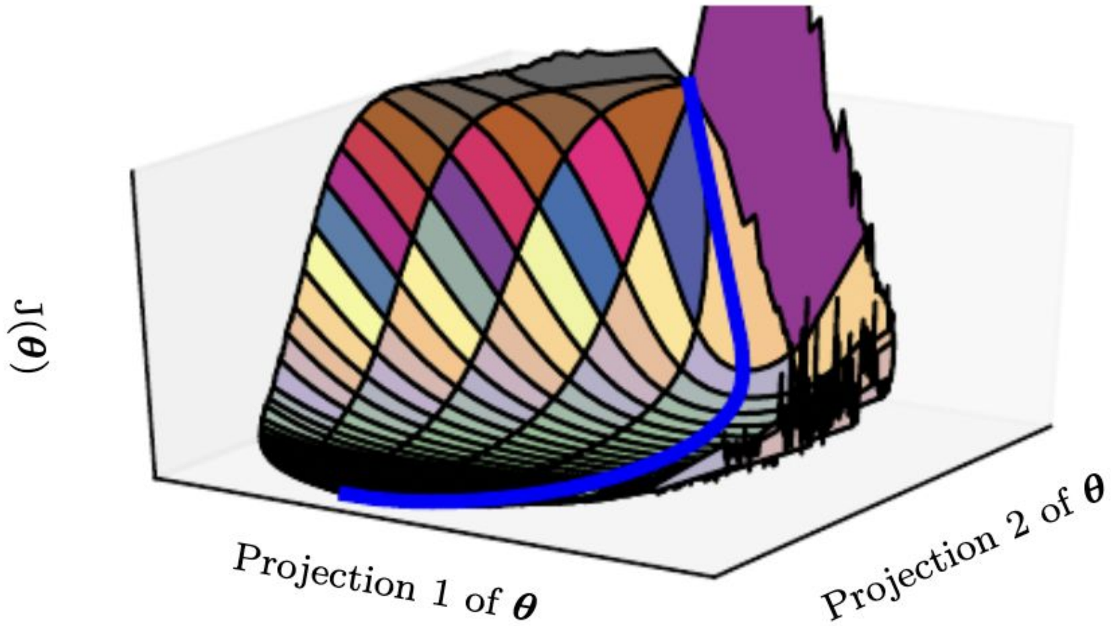
Challenges in neural network optimization

1. Ill-Conditioning



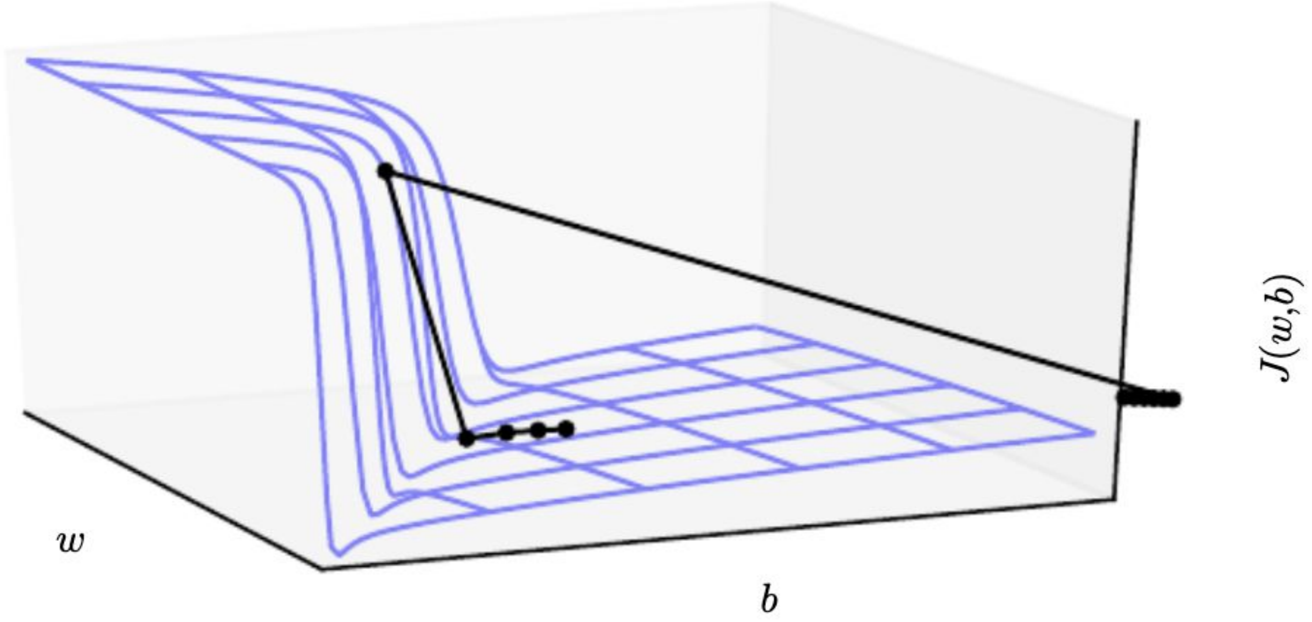
Challenges in neural network optimization

2. Non-convex optimization



Challenges in neural network optimization

3. Cliffs and exploding gradients



Challenges in neural network optimization

4. Initialization of “descent” algorithms