Optimization

\[ \nabla L(\theta, \psi) \]

- Timeline
- Optimization
  - Momentum
  - Adaptive Learning Rates
    (Adagrad, RMSProp, Adam)

Timeline

<table>
<thead>
<tr>
<th>Foundations</th>
<th>Basics</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin Alg</td>
<td>Introductory</td>
<td>MBSGD</td>
<td>GANs</td>
</tr>
<tr>
<td>Cooling</td>
<td>Ethics</td>
<td>Optimization</td>
<td>Hyper tuning</td>
</tr>
<tr>
<td>CLI</td>
<td>Competing techniques</td>
<td>Regularization</td>
<td>Attention</td>
</tr>
<tr>
<td></td>
<td>Single</td>
<td>CNN</td>
<td>Deployed Apps</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>RNN</td>
<td></td>
</tr>
</tbody>
</table>

Don't wait until we cover something in class to use it.
Mini-Batch SGD

\[ \Theta_{t+1} := \Theta_t - \eta \frac{\nabla \mathcal{L}(\hat{y}_b, y_b)}{\nabla \Theta_t} \]

Matrix of all parameters

Learning rate

Loss function gradient w.r.t. each parameter

- How do we pick \( \eta \)?
- Should it be the same for all parameters?
- How do we escape saddle points?

Saddle Points

Momentum

Top-Down View

3D-View
"Exponential Moving Average" of $\theta$

\[
V_{t+1} := B_m V_t + (1 - B_m) \overline{V}_{0:t-1} \\
\theta_{t+1} := \theta_t - \eta \nabla V_{t+1}
\]
RMSProp "Root-Mean-Square Backprop"

\[ g_{t+1} := \beta R g_t + (1 - \beta R) \nabla \theta L_b \]

\[ \Theta_{t+1} := \Theta_t - \eta \frac{\nabla \theta L_b}{\sqrt{g_{t+1}^2 + \varepsilon}} \]

- What if \( g^2 \) is large? \( \rightarrow \) smaller steps
- What if \( g^2 \) is small? \( \rightarrow \) larger steps
Adam "Adaptive Moment Estimation"

- A good first choice
- Combines momentum w/ RMSProp

\[ V_{t+1} = \beta_m V_t + (1 - \beta_m) \theta L_0 \]

\[ \hat{V}_{t+1} = \frac{V_{t+1}}{1 - \beta_m^t} \]

\[ \hat{g}^2_{t+1} = \beta_r \hat{g}^2_t + (1 - \beta_r) \theta L_b^2 \]

\[ \hat{g}^2_{t+1} = \frac{\hat{g}^2_{t+1}}{1 - \beta_r^t} \]

\[ \Theta_{t+1} := \Theta_t - \eta \frac{\hat{V}_{t+1}}{\sqrt{\hat{g}^2_{t+1}} + \epsilon} \]

Bias Correction for EMA

\[ \frac{1}{1 - \beta_m} = 0.9 \]

Learning rate adaptivity