today: finish BCFW

application of BCFW to sumstruct:

\[ \text{getting } s^i \text{ is one loss-augmented decoding call for example } i \]
\[ \lambda^j = \frac{\lambda^j_{t+1}}{\lambda^j_{t}} + \delta \left( s_i \delta^j - \lambda^j_{t+1} \right) \]

\[ w = A \alpha = \sum \lambda^j \frac{1}{A^j} \]

\( \text{(worst case - } O(n^2) \text{)} \)

\[ \text{need before these in memory} \]

\[ \text{store } A^j \text{’s} \]

\( \text{(memory/computation tradeoff)} \)

\( \text{note: for hessian search, also need to store } E \Sigma \alpha_i(\lambda^i) \)

convergence constants:

for sumstruct, can show that

\[ C_{\text{sumstruct}} = \frac{4B_i^2}{\lambda^2} \]

\[ R_i = \max_{i \neq i'} \frac{1}{\lambda^2} \]

Hessian:

\[ \left( \lambda A^j A^j \right)_{i,i'} \]

\[ \max_{i \neq i'} \frac{1}{\lambda^2} \]

\[ \frac{d^2}{d \lambda^2} H \Delta \lambda \]

\[ \max \left< \lambda \Delta \lambda, \lambda \Delta \lambda \right> \frac{1}{\lambda^2} \frac{d^2}{d \lambda^2} \]

\[ \Rightarrow C_{\text{sumstruct}} = \sum_{i=1}^{n} \left< \delta_i, \delta_i \right> \leq \frac{4B_i^2}{\lambda^2} \]

\[ i.e. \text{ BCFW is } “n \text{ times}” \text{ faster than both FW for sumstruct?} \]

importance of affine invariance:

\[ C_{\text{sumstruct}} \leq \| \lambda \|_1 \| \lambda \|_1 \| M \|_2^2 \]
if you use $l_2$-norm, we get a bad bound? \[ \text{diam}(\mathcal{M}) = 2n \]

Lipschitz constant in $l_2$-norm = largest e-value of Hessian

recall Hessian $\nabla^2 f = \frac{1}{n} \left( \left< \nabla f(y), \nabla f(y) \right> \right) \cdot (y, y)

\text{say e.g., } \left< \nabla f(y), \nabla f(y) \right> \text{ is not a l.o.b. of output}

\text{(i.e., } \text{for } i \neq j \text{)}

\Rightarrow \text{get largest e-value can scale with diam of the matrix} \Rightarrow \text{really bad here because}

\text{diam is exponentially}

\text{instead, want to use } l_1 \text{-norm of } \Delta_{12};

\text{i.e., } l_1 \text{-norm for Lipschitz constant}

\text{then get } L = \left( \text{diam}(\Delta_{12}); \right)^2 \approx \frac{4B^2}{\delta}

\underline{variance reduced SGD}

\text{setup: } \min_{x \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} f_i(x)

\text{where } f \text{ is } \mu \text{-strong convex}

\text{L-smooth}

\text{linear rate}

\begin{align*}
\theta_t & = \theta_{t-1} - \gamma_t \sum_{i=1}^{n} f_i(x_t) \\
\theta_t & = \frac{1}{\gamma_t} \sum_{i=1}^{n} f_i(x_t) \\
\theta_t & = \frac{1}{\mu} \sum_{i=1}^{n} f_i(x_t) \\
\end{align*}

\text{batch gradient method}

\text{[Taylor 18th century]}

\text{stochastic gradient method}

\text{aka incremental gradient method}

\text{[Robbins \& Monro 1951]}

\log(\text{sec}) - \theta_t

\text{time}

\text{variance reduced SGD methods?}
SAG (Stochastic average gradient)

1. Compute intercept big surprise, converging slowly and fast
2. Update intercepts
3. Compute weights big surprise, converging slowly and fast
4. Update weights
5. Compute intercepts
6. Update intercepts
7. Compute weights
8. Update weights

Example: l-rg. log regression on RCM

\[
\text{P}_n = \frac{1}{1+e^{-\text{ln}(n)}}, \quad \theta = \text{atan}(\text{ln}(n))
\]

SAG convergence rate

\[
\text{min} = \frac{1}{\text{ln}(n)}
\]

where

\[
\text{min} = \frac{1}{\text{ln}(n)}
\]

Example: l-rg. log regression on RCM

\[
\text{P}_n = \frac{1}{1+e^{-\text{ln}(n)}}, \quad \theta = \text{atan}(\text{ln}(n))
\]

SAG convergence rate

\[
\text{min} = \frac{1}{\text{ln}(n)}
\]

where

\[
\text{min} = \frac{1}{\text{ln}(n)}
\]
practical aspects

- See Schmidt & et al., Math Proj. 2016 paper

a) storage: \( f_i(w) = h(x_i^T w) \Rightarrow \frac{\partial f_i}{\partial w} = h'(x_i^T w) x_i \)
    instead of \( O(n^2) \) storage \( \rightarrow \) \( O(n) \) storage

b) initialization of \( g_i's \)? best: run SGD for one pass then switch SAG/SAGA

c) step size?
   - \( \frac{1}{(4)L} \)
   - Cheap line search heuristic (comes from FISTA)
   - While \( f_i(w_k - \frac{1}{L} \frac{\partial f_i}{\partial w}) > f_i(w_k) - \frac{1}{2}(\frac{1}{L})^2 \|

   \begin{align*}
   &\text{set } L_{new} = 2 L_{old} \\
   \text{else } L_{new} = (1 - \frac{1}{2}) L_{old}
   \end{align*}

d) non-uniform sampling? sample \( i \sim \frac{L_i}{\sum L_i} \)

e) stopping criterion?
you can use \( \frac{1}{n} \sum_j g_j(x) \) as approx \( \nabla f(x) = \frac{1}{n} \sum_j \nabla f_j(x) \)

f) sparse features?
\( x_{t+1} = x_t - \gamma \left[ \nabla f_i(x_t) - \frac{g_i}{\|
\right] \right] \) (sparse SAGA)

weighted projection on support of \( x_t \)
\( s_t = \frac{g_t}{\gamma} u : x_t^u = g_t^u \)