Average Stochastic Gradient Descent for Autoencoder

Sixin Zhang, Clément Farabet, Marco Scoffier, Ross Goroshin, Koray Kavukcuoglu and Yann LeCun, NYU
Plan

Setup

Initialization

Averaging
Plan

Setup

Initialization

Averaging
Autoencoder

+ multi-level perceptron

+ encoder: $h_0 \rightarrow h_L$

+ decoder: $h_L \rightarrow h_{2L}$

+ non-linearity: $tanh$
Challenge

+ $L = 2$

$|h_0| = 3072$, $|h_1| = 6144$, $|h_2| = 100$, $|h_3| = 6144$, $|h_4| = 3072$

+ tied weights

$W_4 = W_1^T$, $W_3 = W_2^T$

+ large scale:

$W1$: $|h_1| \times |h_0| = 6K \times 3K = 18M$

+ goal: minimize the reconstruction mean squared error ($mse$)

$$mse = \frac{1}{2m} \sum_{i=1}^{m} \left\| W_1^T \sigma(W_2^T \sigma(W_2 \sigma(W_1 x_i + b_1) + b_2) + b_3) + b_4 - x_i \right\|^2$$
What we tried ...

- SGD
- Averaged SGD
- Modified Averaged SGD
- Second order SGD - Diagonal approx to Hessian
- Optimal Learning Rate
- Initialization
- Normalize the data and train on a transformed problem
- Predictive Sparse Decomposition (PSD)
- LBFGS, CG
- PCA
What we tried ...

- SGD
- Averaged SGD
- Modified Averaged SGD
- Second order SGD - Diagonal approx to Hessian
- Optimal Learning Rate
- Initialization
- Normalize the data and train on a transformed problem
- Predictive Sparse Decomposition (PSD)
- LBFGS, CG
- PCA
Averaged Stochastic Gradient Descent (ASGD)

**SGD**
(with L2 regularization)

\[
\theta_{t+1} = \theta_t - \eta_t \left( \frac{\partial L_{t+1}(\theta_t)}{\partial \theta} + \lambda \theta_t \right)
\]

\[
\eta_t = \frac{\eta_0}{(1 + \lambda \eta_0 t)^\alpha}
\]

**ASGD**
(t0: starting point)

\[
\bar{\theta}_t = \begin{cases} 
\theta_t & t < t_0 \\
\frac{1}{t-t_0+1} \sum_{k=t_0}^t \theta_k & t \geq t_0 
\end{cases}
\]

**References:**

+ Polyak and Juditsky. *Acceleration of stochastic approximation by averaging*. SIAM. 1992

\[
\alpha = 0.75, \eta_0 = 10^{-4}, \lambda = 10^{-3}
\]
Problems with ASGD

- How to initialize?
- When to start averaging?
Plan

Setup

Initialization

Averaging
Random Initialization of Weights

Idea: keep the variance of each hidden unit invariant across layers, as well as the variance of the back-propagated gradients.

\[
W_k \sim U \left( -\frac{\sqrt{6}}{\sqrt{|h_k| + |h_{k-1}|}}, \frac{\sqrt{6}}{\sqrt{|h_k| + |h_{k-1}|}} \right)
\]

X. Glorot, Y. Bengio, 2010
Activations with Random Initialization

Initial

Final

units

samples
Activations with Random Initialization

The saturation in the mid-layer $h_2$ greatly reduces the capacity of Autoencoder units.
Activations with Random Initialization

Maybe it's caused by the initial "saturation" in $h_2$, so let's center the bias!

Initial

Final

units

samples
Center the **Bias**!

**Idea**: keep the mean of each hidden layer's units zero.

$$mse = \frac{1}{2m} \sum_{i=1}^{m} \left\| W_1^T \sigma(W_2^T \sigma(W_2 \sigma(W_1 x_i + b_1) + b_2) + b_3) + b_4 - x_i \right\|^2$$

\[b_4 = E(h_0)\]
\[b_1 = E(-W_1 h_0 | W_1) = -W_1 b_4\]
\[b_3 = E(h_1 | W_1)\]
\[b_2 = E(-W_2 h_1 | W_2) = -W_2 b_3\]
Random Weights + Bias Centering

Initial

Final
Random Weights + Bias Centering

The saturation of the mid-layer $h_2$ is gone
Random Weights + Bias Centering

however, still some saturation in $h_1$
Our submission

+ Random Weights for W1

\[ W_k \sim U \left( -\frac{\sqrt{6}}{\sqrt{|h_0| + |h_{k-1}|}}, \frac{\sqrt{6}}{\sqrt{|h_0| + |h_{k-1}|}} \right) \]

+ Bias Centering

\[ b_4 = E(h_0) \]
\[ b_1 = E(-W_1 h_0 | W_1) = -W_1 b_4 \]
\[ b_3 = E(h_1 | W_1) \]
\[ b_2 = E(-W_2 h_1 | W_2) = -W_2 b_3 \]

+ PCA Weights for W2

\( W_2 \) corresponds to the 100 first principle components of \( \text{Cov}(h_1, h_1) \)
Our submission ...
Our submission is still problematic..

long-term initialization effect on $h_2$ (caused by big eigenvalues of PCA)
Why still some saturation in final $h_1$?
Untie the weights!

**Intuition:**
- the variance of each layer's unit is determined by the variance of input weights and the fanin (input size)
- fanin of $h_2 = 6144$, fanin of $h_3 = 100$
- when the variance of the inputs is large the same tied weights must both **shrink** the variance in the encoder and **expand** the variance in the decoder

**Proposed Solution:**

$W_4 \neq W_1^T$

$W_3 \neq W_2^T$
the saturation of mid-layer $h_2$ was caused by the tied weights!
but why still some saturation in final h2?
(Random Weights + Bias Centering)

Initial

Tied

Final

Untied

Final

bias centering still helps to avoid saturation!
Center the bias and Untie the weights!
Plan

Setup

Initialization

Averaging
Stochastic Gradient Descent (SGD)

\[ \theta_{t+1} = \theta_t - \eta_t \left( \frac{\partial L_{t+1}(\theta_t)}{\partial \theta} + \lambda \theta_t \right) \]
Averaged SGD (ASGD)

- Weight updates start with SGD
- At time $t_0$ averaging starts; the output is the average of all the weights from $t_0$ to $t$
- SGD continues on the original non-averaged weights

Mathematically:

$$\bar{\theta}_t = \begin{cases} 
\theta_t & t < t_0 \\
\frac{1}{t-t_0+1} \sum_{k=t_0}^{t} \theta_k & t \geq t_0
\end{cases}$$
However, we submitted ...

- Exactly like ASGD, except
- Starting at $t_0 = 12$ epochs, once every $T = 10000$ parameter updates, we replaced the SGD weights by the averaged weights.

**SGD weights replaced by the averaged weights**
When should we start averaging?

Our submission

Choose $t_0$

The later, the better?
What won

- Fast implementation on _GPU_ paralleling function evaluations and parameter updates.
- _Initialization_ of the weights and biases to deal with the saturation.
- _Averaging_ ? _We don't know_ ...
Should we average at all?

Yes, Averaged SGD does help!
When should we start averaging?

When? We don't need to know! Just keep moving (Moving ASGD)

ASGD: $t_0 = 12$

MASGD: $\alpha = 0.9997$
It's time ~

Runtime: LBFGS vs. MASGD

http://www.cs.nyu.edu/~zsx/nips2011