TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

Stochastic Gradient Descent (SGD)

The Classical Convergence Theorem

RMSProp, Momentum and Adam

Scaling Learning Rates with Batch Size

Gradient Flow and Langevin Dynamics

Vanilla SGD

$$\Phi \mathrel{-}= \eta \hat{g}$$

$$\hat{g} = E_{(x,y)\sim\mathrm{Batch}} \nabla_\Phi \operatorname{loss}(\Phi, x, y)$$

$$g = E_{(x,y)\sim\mathrm{Pop}} \nabla_\Phi \operatorname{loss}(\Phi, x, y)$$

Issues

- **Gradient Estimation.** The accuracy of $\hat{g}$ as an estimate of g.
- **Gradient Drift (second order structure).** The fact that g changes as the parameters change.
- **Convergence.** To converge to a local optimum the learning rate must be gradually reduced toward zero.
- **Exploration.** Since deep models are non-convex we need to search over the parameter space. SGD can behave like MCMC.

A One Dimensional Example

Suppose that y is a scalar, and consider

$$\begin{aligned}
\mathrm{loss}(\beta, y) &= \frac{1}{2}(\beta - y)^2 \\
g &= \nabla_\beta \, E_{y\sim\mathrm{Pop}} \, \frac{1}{2}(\beta - y)^2 \\
&= \beta - E_{y\sim\mathrm{Pop}} \, y \\
\hat{g} &= \beta - E_{y\sim\mathrm{Batch}} \, y
\end{aligned}$$

Even if β is optimal, for a finite batch we will have $\hat{g} \neq 0$.

The Classical Convergence Theorem

$$\Phi \mathrel{-}= \eta_t \nabla_\Phi \operatorname{loss}(\Phi, x_t, y_t)$$

For "sufficiently smooth" non-negative loss with

$$\eta_t > 0 \quad \text{and} \quad \lim_{t \to \infty} \eta_t = 0 \quad \text{and} \quad \sum_t \eta_t = \infty,$$

we have that the training loss of Φ converges (in practice Φ converges to a local optimum of training loss).

Rigor Police: One can construct cases where Φ converges to a saddle point or even a limit cycle.

See "Neuro-Dynamic Programming" by Bertsekas and Tsitsiklis proposition 3.5.

Physicist's Proof of the Convergence Theorem

Since $\lim_{t \to 0} \eta_t = 0$ we will eventually get to arbitrarilly small learning rates.

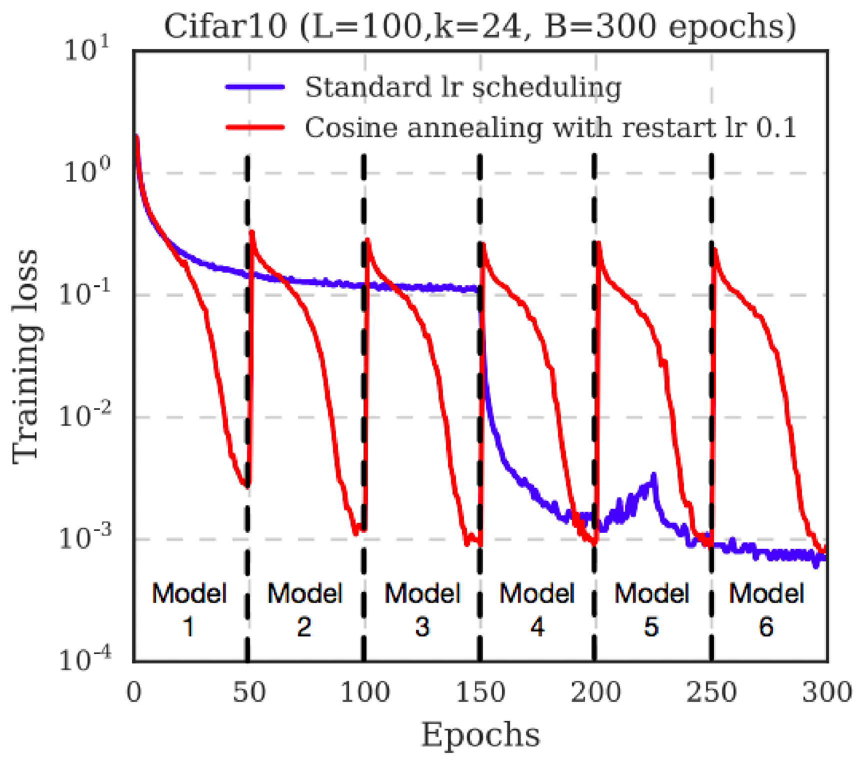
For sufficiently small learning rates any meaningful update of the parameters will be based on an arbitrarily large sample of gradients at essentially the same parameter value.

An arbitrarily large sample will become arbitrarily accurate as an estimate of the full gradient.

But since $\sum_t \eta_t = \infty$, no matter how small the learning rate gets, we still can make arbitrarily large motions in parameter space.

SGD as a form of MCMC

Learning Rate as a Temperature Parameter



Gao Huang et. al., ICLR 2017

Standard Non-Vanilla SGD Algorithms

Digression on Running Averages

Consider a sequence x_1, x_2, x_3, ...

For $t \geq N$, consider the average of the N most recent values.

$$\tilde{\mu} = \frac{1}{N} \sum_{s=t-N+1}^{t} x_s$$

This can be approximated more efficiently with

$$\hat{\mu}_0 = 0$$

$$\hat{\mu}_t = \left(1 - \frac{1}{N}\right) \hat{\mu}_{t-1} + \left(\frac{1}{N}\right) x_t \quad t \geq 1$$

$$\hat{\mu}_t \approx \tilde{\mu}_t \quad t > N$$

Running Averages

More explicitly, for $\hat{\mu}_0 = 0$, the update

$$\hat{\mu}_t = \left(1 - \frac{1}{N}\right)\hat{\mu}_{t-1} + \left(\frac{1}{N}\right)x_t$$

gives

$$\hat{\mu}_t = \frac{1}{N}\sum_{1 \leq s \leq t}\left(1 - \frac{1}{N}\right)^{-(t-s)} x_s$$

where we have

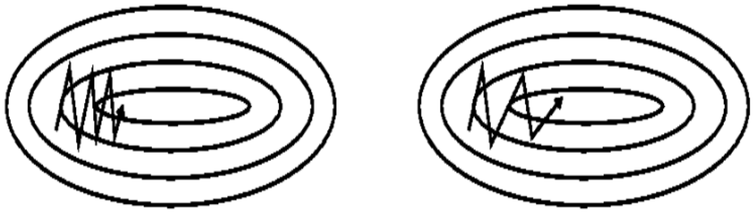
$$\sum_{n \geq 0}\left(1 - \frac{1}{N}\right)^{-n} = N$$

Momentum

$$v_t = \left(1 - \frac{1}{N_g}\right) v_{t-1} + \eta * \hat{g}_t \quad N_g \text{ is typically 10 or 100}$$

$$\Phi_{t+1} = \Phi_t - v_t$$

The theory of momentum is generally given in terms of second order structure.



Rudin's blog

Momentum

However, second order analyses are controversial in Deep Learning.

We can perhaps get insight by reparameterizing the momentum equations.

We are Entering a Twilight Zone (TZ)

The twilight zone is material for which I do not know of a reference.

Conjugacy of SGD Hyperparameters

For an objective function $f(x, y)$ we say that x and y are conjugate if changing x does effect the loss gradient for y.

In other words if

$$\frac{\partial f(x, y)}{\partial x \partial y} = 0$$

We are seeking a nonlinear transformation on hyperparameters yielding robust conjugacy.

Solving for η in terms of N_g (TZ)

Let η' be a learning rate that is appropriate without momentum ($N_g = 1$).

We will solve for a learning rate η as a function of N_g so as to minimize the perturbation of SGD as we increase N_g.

An Equivalent Formulation of Momentum

$$\begin{aligned}
v_t &= \left(1 - \frac{1}{N_g}\right) v_{t-1} + \eta \hat{g}_t \\
&= \left(1 - \frac{1}{N_g}\right) v_{t-1} + \frac{1}{N_g}(N_g \eta \hat{g}_t) \\
v_t &= N_g \eta \mu_t \;\; \text{for } \mu_t = \left(1 - \frac{1}{N_g}\right) \mu_{t-1} + \frac{1}{N_g} \hat{g}_t \\
\Phi_{t+1} &= \Phi_t - v_t \\
\Phi_{t+1} &= \Phi_t - N_g \eta \mu_t
\end{aligned}$$

Solving for η (TZ)

$$\mu_t = \left(1 - \frac{1}{N_g}\right)\mu_{t-1} + \frac{1}{N_g}\hat{g}_t$$

$$\text{we have } \Phi_{t+1} = \Phi_t - N_g\eta\,\mu_t$$

$$\text{unperturbed is } \Phi_{t+1} = \Phi_t - \eta'\mu_t$$

This gives $\eta = \eta'/N_g$.

We expect that that η' and N_g exhibit more independence (conjugacy) than η, N_g.

Scaling η and N_g with Batch Size

Recent work has show that scaling hyper-parameters with the batch size can lead to effective learning with very large (highly parallel) batches.

Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour, Goyal et al., 2017.

Don't Decay the Learning Rate, Increase the Batch Size, Smith et al., 2018

Solving for η as a Function of B

Let η' be a learning rate appropriate for $N_g = 1$ and $B = 1$.

We will solve for η as a function of η' and B so as to minimize the perturbation of SGD as we increase B.

Solving for η as a function of B

Consider two consecutive updates for a batch size of 1 with learning rate η'.

$$\begin{aligned}
\Phi_{t+1} &= \Phi_t - \eta' \nabla_\Phi \mathrm{loss}(\Phi_t, x_t, y_t) \\
\Phi_{t+2} &= \Phi_{t+1} - \eta' \nabla_\Phi \mathrm{loss}(\Phi_{t+1}, x_{t+1}, y_{t+1}) \\
&\approx \Phi_{t+1} - \eta' \nabla_\Phi \mathrm{loss}(\Phi_t, x_{t+1}, y_{t+1}) \\
&= \Phi_t - \eta'((\nabla_\Phi \mathrm{loss}(\Phi_t, x_t, y_t)) + (\nabla_\Phi \mathrm{loss}(\Phi_t, x_{t+1}, y_{t+1})))
\end{aligned}$$

Solving for η as a function of B

Let η be the learning rate for batch size B.

$$\Phi_{t+2} \approx \Phi_t - \eta'((\nabla_\Phi \text{loss}(\Phi_t, x_t, y_t)) + (\nabla_\Phi \text{loss}(\Phi_t, x_{t+1}, y_{t+1})))$$

$$= \Phi_t - 2\eta'\ \hat{g} \quad \text{for} \quad B = 2$$

We have $\Phi \mathrel{\texttt{-=}} \eta\ \hat{g}$

unperturbed is $\Phi \mathrel{\texttt{-=}} B\eta'\ \hat{g}$

This gives $\eta = B\eta'$.

We expect conjugacy for the parameters η' and B.

Solving for N_g as a Function of B

Let N'_g be a momentum parameter appropriate for $B = 1$.

We will solve for N_g as a function of B so as to minimize the perturbation of SGD as we increase B.

For batch size B, $\hat{\mu}_t$ is an average of $\hat{g}$ over $N_g B$ gradient values.

SGD seems minimally perturbed by

$$N_g B = N'_g \quad \text{or} \quad N_g = N'_g / B$$

We expect conjugacy for the parameters N'_g and B.

Putting It Together (TZ)

$$\begin{aligned} N_g &= \min(1, N_g'/B) \\ \eta &= B\eta'/N_g \end{aligned}$$

We expect conjugacy for η', N_g' and B.

RMSProp

RMSProp is based on a running average of $\hat{g}[i]^2$ for each real-valued model parameter i.

$$s_t[i] \;=\; \left(1 - \frac{1}{N_s}\right) s_{t-1}[i] + \frac{1}{N_s}\hat{g}_t[i]^2 \quad N_s \text{ typically 100 or 1000}$$

$$\Phi_{t+1}[i] \;=\; \Phi_t[i] - \frac{\eta}{\sqrt{s_t[i] + \epsilon}}\;\hat{g}_t[i]$$

RMSProp

$$s_t[i] = \left(1 - \frac{1}{N_s}\right) s_{t-1}[i] + \frac{1}{N_s}\hat{g}_t[i]^2 \quad N_s \text{ typically 100 or 1000}$$

$$\Phi_{t+1}[i] = \Phi_t[i] - \frac{\eta}{\sqrt{s_t[i] + \epsilon}} \; \hat{g}_t[i]$$

For $B = 1$ we have $s[i] \approx E\; \hat{g}[i]^2 = g[i]^2 + \sigma[i]^2$

We should expect $\sigma[i] >> g[i]$.

We can therefore think of RMSProp as $\Phi[i]$ -= $\eta(\hat{g}[i]/\sigma[i])$

RMSProp is Theoretically Mysterious

$$\Phi[i] \mathrel{-}= \eta\, \frac{\hat{g}[i]}{\sigma[i]} \quad (1) \qquad\qquad \Phi[i] \mathrel{-}= \eta\, \frac{\hat{g}[i]}{\sigma^2[i]} \quad (2)$$

Although (1) seems to work better, (2) is better motivated theoretically. To see this we can consider units.

If parameters have units of “weight”, and loss is in bits, then (2) type checks with η having units of bits — the numerical value of η has no dependence on the choice of the weight unit.

Consistent with the dimensional analysis, many theoretical analyses support (2) over (1) contrary to apparent empirical performance.

Scaling η with Batch Size for RMSProp (TZ)

Current RMSProp framework implementations do not scale well with batch size.

The meaning of $\hat{g}_i^2$ changes when increasing B.

$\hat{g}_i$ is an average over a batch — averaging reduces variance.

$$E\ \hat{g}_i^2 = = \mu_i^2 + \frac{\sigma_i^2}{B}$$

Scaling η with Batch Size for RMSProp (TZ)

$$E\ \hat{g}_i^2 = \mu_i^2 + \frac{\sigma_i^2}{B}$$

To preserve the behavior of RMSProp while increasing B we need a running average of $g_{b,i}^2$ for individual batch elements b.

This requires augmenting backpropagation to compute an average (or sum) of $g_{b,i}^2$ as well as an average of $g_{b,i}$ within each batch.

Adam — Adaptive Momentum

Adam combines momentum and RMSProp.

It also uses “bias correction” of running averages.

A Digression on Bias Correction of Running Averages

Consider a sequence $x_1,\ x_2,\ x_3, \ldots$ and consider the following for N large.

$$\begin{aligned}\hat{\mu}_0 &= 0\\ \hat{\mu}_t &= \left(1 - \frac{1}{N}\right)\hat{\mu}_{t-1} + \left(\frac{1}{N}\right)x_t\end{aligned}$$

For $\mu \doteq E\ x$ we have $E\ \hat{\mu}_1 = \mu/N$.

For $t << N$ we have $E\ \hat{\mu}_t \approx (t/N)\mu$.

Bias Correction of Running Averages

The following running average maintains the invariant that $\hat{\mu}_t$ is exactly the average of $x_1, \ldots, x_t$.

$$\begin{aligned}
\hat{\mu}_0 &= 0 \\
\hat{\mu}_t &= \left(\frac{t-1}{t}\right)\hat{\mu}_{t-1} + \left(\frac{1}{t}\right)x_t \\
&= \left(1 - \frac{1}{t}\right)\hat{\mu}_{t-1} + \left(\frac{1}{t}\right)x_t
\end{aligned}$$

But this fails to track a moving average for $t >> N$.

Bias Correction of Running Averages

The following avoids the initial bias toward zero while still tracking a moving average.

$$\begin{aligned}\hat{\mu}_0 &= 0 \\ \hat{\mu}_t &= \left(1 - \frac{1}{\min(N,t)}\right)\hat{\mu}_{t-1} + \left(\frac{1}{\min(N,t)}\right)x_t\end{aligned}$$

The published version of Adam has a more obscure form of bias correction which yields essentially the same effect.

Adam (simplified)

$$\begin{aligned}
\mu_0[i] &= s_0[i] = 0 \\
\mu_t[i] &= \left(1 - \frac{1}{\min(t, N_g)}\right) \mu_{t-1}[i] + \frac{1}{\min(t, N_g)} \hat{g}_t[i] \\
s_t[i] &= \left(1 - \frac{1}{\min(t, N_s)}\right) s_{t-1}[i] + \frac{1}{\min(t, N_s)} \hat{g}_t[i]^2 \\
\Phi_{t+1}[i] &= \Phi_t - \frac{\eta}{\sqrt{s_t[i] + \epsilon}}\ \mu_t[i]
\end{aligned}$$

Gradient Flow

Gradient flow is a non-stochastic (deterministic) model of stochastic gradient descent (SGD).

Gradient flow is defined by the total gradient differential equation

$$\frac{d\Phi}{dt} = -g(\Phi) \qquad g(\Phi) = \nabla_\Phi \; E_{(x,y)\sim \mathrm{Train}} \; \mathcal{L}(\Phi, x, y)$$

We let $\Phi(t)$ be the solution satisfying $\Phi(0) = \Phi_{\mathrm{init}}$.

Gradient Flow

$$\frac{d\Phi}{dt} = -g(\Phi)$$

For small values of Δt this differential equation can be approximated by

$$\Delta\Phi = -g(\Phi)\Delta t$$

Here Δt can be interpreted as a learning rate.

Gradient Flow also Models SGD

Consider the SGD update with $\hat{g}$ from a random data point

$$\Delta\Phi = -\hat{g}\Delta t$$

This also converges to deterministic gradient flow as Δt goes to zero.

To see this note that we can divide the updates into $\sqrt{N}$ blocks each of size $\sqrt{N}$. The "time" spent within each block is $t/\sqrt{N}$ which converges to 0. But the number of updates within each block grows as $\sqrt{N}$ and hence the average update within a block converges to g.

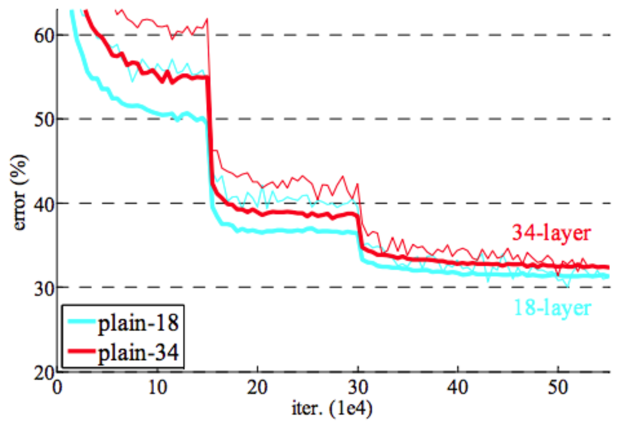
Gradient Flow also Models SGD

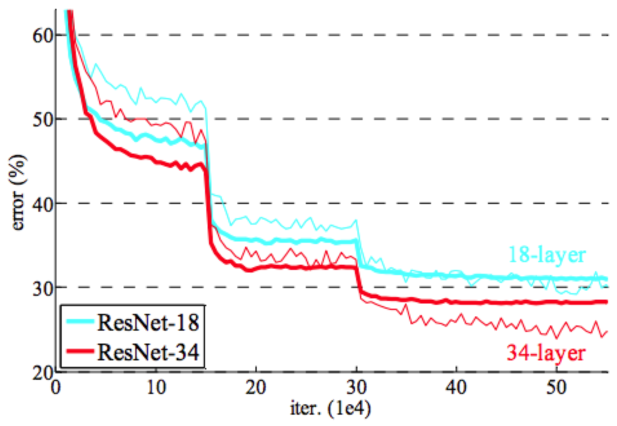
Consider the SGD update with $\hat{g}$ from a random data point

$$\Delta\Phi = -\hat{g}\Delta t$$

The sum of the learning rates can be interpreted as “time”.

MCMC models of SGD



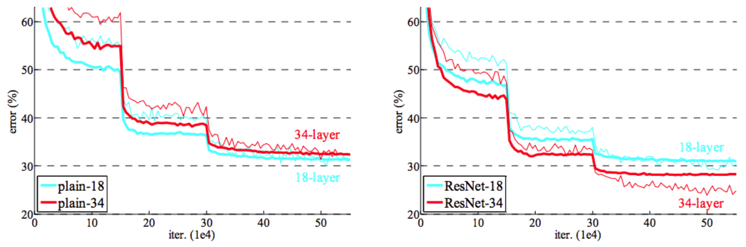


These Plots are from the original ResNet paper. Left plot is for CNNs without residual skip connections, the right plot is ResNet.

Thin lines are training error, thick lines are validataion error.

In all cases η is reduced twice, each time by a factor of 2.

Converged Loss as a Function of η



For each value of η we converge at a loss $\mathcal{L}(\eta)$.

$$\begin{aligned} \mathcal{L}(0) &\doteq \lim_{\eta \to 0} \mathcal{L}(\eta) \\ &= \mathcal{L}(\Phi^*) \quad \Phi^* \text{ a local optimum} \end{aligned}$$

Can we do a Taylor expansion of $\mathcal{L}(\eta)$?

$$\mathcal{L}(\eta) = \mathcal{L}(\Phi^*) + \left(\left. \frac{d\mathcal{L}}{d\eta} \right|_{\eta=0} \right) \eta + \ldots$$

A Nice Theorem (TZ)

$$\mathcal{L}(\eta) = \mathcal{L}(\Phi^*) + \left(\left. \frac{d\mathcal{L}}{d\eta} \right|_{\eta=0} \right) \eta + \ldots$$

Let i index a training example and let g_i denote $\nabla_\Phi \mathcal{L}_i(\Phi)$ at $\Phi = \Phi^*$.

Theorem:

$$\left. \frac{\partial \mathcal{L}(\eta)}{\partial \eta} \right|_{\eta=0} \quad = \quad \frac{1}{4}\, E_i\, ||g_i||^2$$

Proof Step 1 (TZ)

Let i index a training example and let $\mathcal{L}_i(\Phi^* + \Delta\Phi)$ be the loss on training example i with model parameters $\Phi^* + \Delta\Phi$.

We take a second order Taylor expansion.

$$\mathcal{L}(\Phi) = E_i\ \mathcal{L}_i(\Phi)$$

$$\mathcal{L}_i(\Phi^* + \Delta\Phi) = \mathcal{L}_i + g_i\Delta\Phi + \frac{1}{2}\Delta\Phi^\top H_i \Delta\Phi$$

$$E_i\ g_i = 0$$

$$E_i\ H_i \quad \text{is positive definite}$$

Proof: Step 2 (TZ)

$$
\begin{aligned}
\mathcal{L}(\eta) &= \quad E_{\Delta\Phi\sim P_\eta}\, E_i \quad \mathcal{L}_i + g_i \Delta_\Phi + \frac{1}{2}\,\Delta\Phi^\top H_i \Delta\Phi \\
&= E_i\, \mathcal{L}_i + E_{\Delta\Phi\sim P_\eta} \quad (E_i\; g_i)\,\Delta\Phi + \frac{1}{2}\,\Delta\Phi^\top (E_i\; H_i)\Delta\Phi \\
&= \mathcal{L}(\Phi^*) \;\; + \;\; E_{\Delta\Phi\sim P_\eta} \;\; \frac{1}{2}\,\Delta\Phi^\top (E_i\; H_i)\Delta\Phi
\end{aligned}
$$

Proof: Step 3 (TZ)

Because P_η is a stationary distribution we must have

$$E_{\Delta\Phi\sim P_\eta} E_i \; ||\Delta\Phi - \eta(g_i + H_i\Delta\Phi)||^2 = E_{\Delta\Phi\sim P_\eta} \; ||\Delta\Phi||^2$$

$$E_{\Delta\Phi\sim P_\eta} E_i \; - 2\eta\Delta\Phi^\top(g_i + H_i\Delta\Phi) + \eta^2||(g_i + H_i\Delta\Phi)||^2 = 0$$

$$E_{\Delta\Phi\sim P_\eta} \; \left(\frac{1}{2}\Delta\Phi^\top(E_i \; H_i)\Delta\Phi\right) = \frac{\eta}{4} \; E_{\Delta\Phi\sim P_\eta} E_i \; ||(g_i + H_i\Delta\Phi)||^2$$

$$\mathcal{L}(\eta) = \mathcal{L}(\Phi^*) + \frac{\eta}{4} \; E_{\Delta\Phi\sim P_\eta} E_i \; ||(g_i + H_i\Delta\Phi)||^2$$

Proof Step 4 (TZ)

$$\mathcal{L}(\eta) = \mathcal{L}(\Phi^*) + \frac{\eta}{4} E_{\Delta\Phi \sim P_\eta} E_i \, ||(g_i + H_i \Delta\Phi)||^2$$

$$\left.\frac{\partial \mathcal{L}(\eta)}{\partial \eta}\right|_{\eta=0} = \frac{1}{4} \lim_{\eta \to 0} \, E_{\Delta\Phi \sim P_\eta} \, E_i \, ||(g_i + H_i \Delta\Phi)||^2$$

$$= \frac{1}{4} \, E_i \, ||g_i||^2$$

Langevin Dynamics

Can we analytically solve for stationary distributions?

Is the stationary distributin some kind of Gibbs Distribution?

Langevin dynamics models both the stationary distribution and non-stationary stochastic dynamics with a continuous time stochastic differential equation.

Langevin Dynaimcs

Consider SGD with $B = 1$.

$$\Phi \mathrel{-}= \eta\hat{g}$$

For N steps of SGD we define $\Delta t = N\eta$.

In Langevin dynamics we hold $\eta > 0$ fixed.

We then consider Δt large compared to η (so that it corresponds to many SGD updates) but small enough so that the gradient distribution does not change during the interval Δt.

Langevin Dynamics

If the mean gradient $g(\Phi)$ is approximately constant over the interval $\Delta t = N\eta$ we have

$$\Phi(t + \Delta t) \approx \Phi(t) - g(\Phi)\Delta t + \eta \sum_{j=1}^{N} (g(\Phi) - \hat{g}_i)$$

The Random variables in the last term have zero mean.

By the law of large numbers a sum (not the average) of N random vectors will approximate a Gaussian distribution where the standard deviation grows like $\sqrt{N}$.

Langevin Dynamics

Let Σ be the covariance matrix of the random variable $\hat{g}$ and assume this is approximately constant over the interval Δt. Let ϵ be a zero mean Gaussian random variable with the same covariance matrix Σ.

$$\begin{aligned}
\Phi(t+\Delta t) &\approx \Phi(t) - g(\Phi)\Delta t + \eta \sum_{j=1}^{N} (g(\Phi) - \hat{g}_i) \\
&\approx \Phi(t) - g(\Phi)\Delta t + \eta\epsilon\sqrt{N} \\
&= \Phi(t) - g(\Phi)\Delta t + \eta\epsilon\sqrt{\frac{\Delta t}{\eta}}
\end{aligned}$$

Langevin Dynamics

$$\begin{aligned}\Phi(t+\Delta t) &\approx \Phi(t) - g(\Phi)\Delta t + \epsilon\sqrt{\eta \Delta t} \qquad \epsilon \sim \mathcal{N}(0,\Sigma) \\ &= \Phi(t) - g(\Phi)\Delta t + \epsilon\sqrt{\Delta t} \qquad \epsilon \sim \mathcal{N}(0,\eta\Sigma)\end{aligned}$$

This can be modeled by a continuous time stochastic process — Langevin dynamics — defined by the notation

$$d\Phi = -g(\Phi)dt + \epsilon\sqrt{dt} \qquad \epsilon \sim \mathcal{N}(0,\eta\Sigma)$$

This is a stochastic differential equation.

If $g(\Phi) = 0$ and $\Sigma = I$ we get Brownian motion.

Langevin Dynamics

$$\Phi(t+\Delta t) \;\approx\; \Phi(t) - g(\Phi)\Delta t + \epsilon\sqrt{\Delta t} \qquad \epsilon \sim \mathcal{N}(0, \eta\Sigma)$$

Note that for $\eta \to 0$ the noise term vanishes. If we then take $\Delta t \to 0$ (at a slower rate) we are back to gradient flow.

In Langevin dynamics we hold $\eta > 0$ fixed.

Stationary Distributions

SGD at $B = 1$ defines a Markov process

$$\Phi \mathrel{-}= \eta \hat{g}$$

Under Langevin dynamics the stationary distribution is a continuous density in parameter space.

If the covariance matrix is isotropic (all eigenvalues are the same) we get a Gibbs distribution.

The 1-D Langevin Stationary Distribution

Consider SGD on a single parameter.

Let p be a probability density on x.

Assume that the gradient $\hat{g}$ has variance σ everywhere.

There is a diffusion flow proportional to $\eta^2\sigma^2 \, dp/dx$.

There is a gradient flow equal to $\eta p \, d\mathcal{L}/dx$.

For a stationary distribution the two flows cancel giving.

$$\alpha\eta^2\sigma^2\frac{dp}{dx} = -\eta p\frac{d\mathcal{L}}{dx}$$

The 1-D Langevin Stationary Distribution

$$\begin{aligned} \alpha\eta^2\sigma^2\frac{dp}{dx} &= -\eta p\frac{d\mathcal{L}}{dx} \\ \frac{dp}{p} &= \frac{-d\mathcal{L}}{\alpha\eta\sigma^2} \\ \ln p &= \frac{-\mathcal{L}}{\alpha\eta\sigma^2} + C \\ p(x) &= \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x)}{\alpha\eta\sigma^2}\right) \quad \alpha \approx 1/10 \end{aligned}$$

We get a Gibbs distribution!

A 2-D Langevin Stationary Distribution

Let p be a probability density on two parameters (x, y).

We consider the case where x and y are completely independent with

$$\mathcal{L}(x, y) = \mathcal{L}(x) + \mathcal{L}(y)$$

For completely independent variables we have

$$\begin{aligned} p(x, y) &= p(x)p(y) \\ &= \frac{1}{Z} \exp\left(\frac{-\mathcal{L}(x)}{\alpha\eta\sigma_x^2} + \frac{-\mathcal{L}(y)}{\alpha\eta\sigma_y^2} \right) \end{aligned}$$

A 2-D Langevin Stationary Distribution

$$
\begin{aligned}
p(x,y) &= \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x)}{\alpha\eta\sigma_x^2} + \frac{-\mathcal{L}(y)}{\alpha\eta\sigma_y^2}\right) \\
&= \frac{1}{Z}\exp\big(-\beta_x\mathcal{L}(x) - \beta_y\mathcal{L}(y)\big)
\end{aligned}
$$

This is not a Gibbs distribution!

It has two different temperature parameters!

Langevin and RMSProp

Suppose we use parameter-specific learning rates η_x and η_y

$$p(x,y) = \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x)}{\alpha\eta_x\sigma_x^2} + \frac{-\mathcal{L}(y)}{\alpha\eta_y\sigma_y^2}\right)$$

Setting $\eta_x = \eta'/\sigma_x^2$ and $\eta_y = \eta'/\sigma_y^2$ gives

$$\begin{aligned} p(x,y) &= \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x)}{\alpha\eta'} + \frac{-\mathcal{L}(y)}{\alpha\eta'}\right) \\ &= \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x,y)}{\alpha\eta'}\right) \quad \text{Gibbs!} \end{aligned}$$

Langevin and RMSProp

Suppose we use parameter-specific learning rates η_x and η_y
Setting $\eta_x = \eta'/\sigma_x^2$ and $\eta_y = \eta'/\sigma_y^2$ gives

$$p(x,y) = \frac{1}{Z}\exp\left(\frac{-\mathcal{L}(x,y)}{\alpha\eta'}\right) \quad \text{Gibbs!}$$

RMSProp sets $\eta_x = \eta'/\sigma_x$ rather than $\eta_x = \eta'/\sigma_x^2$. Empirically RMSProp seems better that the more theoretically motivated algorithm.

A Second Order Algorithm (TZ)

We will derive a learning rate by maximizing a lower bound on the rate of reduction in training loss.

We must consider

- **Gradient Estimation.** The accuracy of $\hat{g}$ as an estimate of g.

- **Gradient Drift (second order structure).** The fact that g changes as the parameters change.

Analysis Plan

We will calculate a batch size B^* and learning rate η^* by optimizing an improvement guarantee for a single batch update.

We then use learning rate scaling to derive the learning rate η_B for a batch size $B << B^*$.

Deriving Learning Rates

If we can calculate B^* and η^* for optimal loss reduction in a single batch we can calculate η_B.

$$\eta_B = B\ \eta_1$$

$$\eta^* = B^* \eta_1$$

$$\eta_1 = \frac{\eta^*}{B^*}$$

$$\eta_B = \frac{B}{B^*}\ \eta^*$$

Calculating B^* and η^* in One Dimension

We will first calculate values B^* and η^* by optimizing the loss reduction over a single batch update in one dimension.

$$g = \hat{g} \pm \frac{2\hat{\sigma}}{\sqrt{B}}$$

$$\hat{\sigma} = \sqrt{E_{(x,y)\sim\text{Batch}} \left(\frac{d\ \text{loss}(\beta, x, y)}{d\beta} - \hat{g} \right)^2}$$

The Second Derivative of $\mathrm{loss}(\beta)$

$$\mathrm{loss}(\beta) \;=\; E_{(x,y)\sim \mathrm{Train}}\; \mathrm{loss}(\beta, x, y)$$

$$d^2\mathrm{loss}(\beta)/d\beta^2 \;\leq\; L \quad \text{(Assumption)}$$

$$\mathrm{loss}(\beta - \Delta\beta) \;\leq\; \mathrm{loss}(\beta) - g\Delta\beta + \frac{1}{2}L\Delta\beta^2$$

$$\mathrm{loss}(\beta - \eta\hat{g}) \;\leq\; \mathrm{loss}(\beta) - g(\eta\hat{g}) + \frac{1}{2}L(\eta\hat{g})^2$$

A Progress Guarantee

$$\begin{aligned}
\mathrm{loss}(\beta - \eta\hat{g}) &\leq \mathrm{loss}(\beta) - g(\eta\hat{g}) + \frac{1}{2}L(\eta\hat{g})^2 \\
&= \mathrm{loss}(\beta) - \eta(\hat{g} - (\hat{g} - g))\hat{g} + \frac{1}{2}L\eta^2\hat{g}^2 \\
&\leq \mathrm{loss}(\beta) - \eta\left(\hat{g} - \frac{2\hat{\sigma}}{\sqrt{B}}\right)\hat{g} + \frac{1}{2}L\eta^2\hat{g}^2
\end{aligned}$$

Optimizing B and η

$$\mathrm{loss}(\beta - \eta\hat{g}) \leq \mathrm{loss}(\beta) - \eta\left(\hat{g} - \frac{2\hat{\sigma}}{\sqrt{B}}\right)\hat{g} + \frac{1}{2}L\eta^2\hat{g}^2$$

We optimize progress per gradient calculation by optimizing the right hand side divided by B. The derivation at the end of the slides gives

$$B^* = \frac{16\hat{\sigma}^2}{\hat{g}^2}, \qquad \eta^* = \frac{1}{2L}$$

$$\eta_B = \frac{B}{B^*}\eta^* = \frac{B\hat{g}^2}{32\hat{\sigma}^2 L}$$

Recall this is all just in one dimension.

Estimating $\hat{g}_{B^*}$ and $\hat{\sigma}_{B^*}$

$$\eta_B = \frac{B\hat{g}^2}{32\hat{\sigma}^2 L}$$

We are left with the problem that $\hat{g}$ and $\hat{\sigma}$ are defined in terms of batch size $B^* >> B$.

We can estimate $\hat{g}_{B^*}$ and $\hat{\sigma}_{B^*}$ using a running average with a time constant corresponding to B^*.

Estimating $\hat{g}_{B^*}$

$$\hat{g}_{B^*} = \frac{1}{B^*} \sum_{(x,y)\sim\text{Batch}(B^*)} \frac{d\ \text{Loss}(\beta, x, y)}{d\beta}$$

$$= \frac{1}{N} \sum_{s=t-N+1}^{t} \hat{g}^s \quad \text{with } N = \frac{B^*}{B} \text{ for batch size } B$$

$$\tilde{g}^{t+1} = \left(1 - \frac{B}{B^*}\right) \tilde{g}^t + \frac{B}{B^*} \hat{g}^{t+1}$$

We are still working in just one dimension.

A Complete Calculation of η (in One Dimension)

$$\begin{aligned}
\tilde{g}^{t+1} &= \left(1 - \frac{B}{B^*(t)}\right)\tilde{g}^{t} + \frac{B}{B^*(t)}\hat{g}^{t+1} \\
\tilde{s}^{t+1} &= \left(1 - \frac{B}{B^*(t)}\right)\tilde{s}^{t} + \frac{B}{B^*(t)}(\hat{g}^{t+1})^2 \\
\tilde{\sigma}^{t} &= \sqrt{\tilde{s}^{t} - (\tilde{g}^{t})^2} \\
B^*(t) &= \begin{cases} K & \text{for } \ t \leq K \\ 16(\tilde{\sigma}^{t})^2/((\tilde{g}^{t})^2 + \epsilon) & \text{otherwise} \end{cases}
\end{aligned}$$

A Complete Calculation of η (in One Dimension)

$$\eta^t = \begin{cases} 0 & \text{for } \; t \leq K \\ \frac{(\tilde{g}^t)^2}{32(\tilde{\sigma}^t)^2 L} & \text{otherwise} \end{cases}$$

As $t \to \infty$ we expect $\tilde{g}^t \to 0$ and $\tilde{\sigma}^t \to \sigma > 0$ which implies $\eta^t \to 0$.

The High Dimensional Case

So far we have been considering just one dimension.

We now propose treating each dimension $\Phi[i]$ of a high dimensional parameter vector Φ independently using the one dimensional analysis.

We can calculate $B^*[i]$ and $\eta^*[i]$ **for each individual parameter** $\Phi[i]$.

Of course the actual batch size B will be the same for all parameters.

A Complete Algorithm

$$\begin{aligned}
\tilde{g}^{t+1}[i] &= \left(1 - \frac{B}{B^*(t)[i]}\right)\tilde{g}^{t}[i] + \frac{B}{B^*(t)[i]}\hat{g}^{t+1}[i] \\
\tilde{s}^{t+1}[i] &= \left(1 - \frac{B}{B^*(t)[i]}\right)\tilde{s}^{t}[i] + \frac{B}{B^*(t)[i]}\hat{g}^{t+1}[i]^2 \\
\tilde{\sigma}^{t}[i] &= \sqrt{\tilde{s}^{t}[i] - \tilde{g}^{t}[i]^2} \\
B^*(t)[i] &= \begin{cases} K & \text{for } \; t \leq K \\ \lambda_B \tilde{\sigma}^t[i]^2/(\tilde{g}^t[i]^2 + \epsilon) & \text{otherwise} \end{cases}
\end{aligned}$$

A Complete Algorithm

$$\eta^t[i] = \begin{cases} 0 & \text{for} \;\; t \leq K \\ \frac{\lambda_\eta \tilde{g}^t[i]^2}{\tilde{\sigma}^t[i]^2} & \text{otherwise} \end{cases}$$

$$\Phi^{t+1}[i] = \Phi^t[i] - \eta^t[i]\hat{g}^t[i]$$

Here we have meta-parameters K, λ_B, ϵ and λ_η.

Appendix: Optimizing B and η

$$\mathrm{loss}(\beta - \eta\hat{g}) \leq \mathrm{loss}(\beta) - \eta\hat{g}\left(\hat{g} - \frac{2\hat{\sigma}}{\sqrt{B}}\right) + \frac{1}{2}L\eta^2\hat{g}^2$$

Optimizing η we get

$$\hat{g}\left(\hat{g} - \frac{2\hat{\sigma}}{\sqrt{B}}\right) = L\eta\hat{g}^2$$

$$\eta^*(B) = \frac{1}{L}\left(1 - \frac{2\hat{\sigma}}{\hat{g}\sqrt{B}}\right)$$

Inserting this into the guarantee gives

$$\mathrm{loss}(\Phi - \eta\hat{g}) \leq \mathrm{loss}(\Phi) - \frac{L}{2}\eta^*(B)^2\hat{g}^2$$

Optimizing B

Optimizing progress per sample, or maximizing $\eta^*(B)^2/B$, we get

$$\begin{aligned}
\frac{\eta^*(B)^2}{B} &= \frac{1}{L^2}\left(\frac{1}{\sqrt{B}} - \frac{2\hat{\sigma}}{\hat{g}B}\right)^2 \\
0 &= -\frac{1}{2}B^{-\frac{3}{2}} + \frac{2\hat{\sigma}}{\hat{g}}B^{-2} \\
B^* &= \frac{16\hat{\sigma}^2}{\hat{g}^2} \\
\eta^*(B^*) = \eta^* &= \frac{1}{2L}
\end{aligned}$$

Appendix II: A Formal Bound for the Vector Case

We will prove that minibatch SGD for a **sufficiently large batch size** (for gradient estimation) and a **sufficient small learning rate** (to avoid gradient drift) is guaranteed (with high probability) to reduce the loss.

This guarantee has two main requirements.

- A smoothness condition to limit gradient drift.
- A bound on the gradient norm allowing high confidence gradient estimation.

Smoothness: The Hessian

We can make a second order approximation to the loss.

$$\ell(\Phi + \Delta\Phi) \approx \ell(\Phi) + g^\top \Delta\Phi + \frac{1}{2}\Delta\Phi^\top H \Delta\Phi$$

$$g = \nabla_\Phi \, \ell(\Phi)$$

$$H = \nabla_\Phi \nabla_\Phi \, \ell(\Phi)$$

The Smoothness Condition

We will assume

$$||H\Delta\Phi|| \leq L||\Delta\Phi||$$

We now have

$$\Delta\Phi^\top H \Delta\Phi \leq L||\Delta\Phi||^2$$

Using the second order mean value theorem one can prove

$$\ell(\Phi + \Delta\Phi) \;\leq\; \ell(\Phi) + g^\top \Delta\Phi + \frac{1}{2}L||\Delta\Phi||^2$$

A Concentration Inequality for Gradient Estimation

Consider a vector mean estimator where the vectors g_n are drawn IID.

$$g_n = \nabla_\Phi \ell_n(\Phi) \qquad \hat{g} = \frac{1}{k}\sum_{n=1}^{k} g_n \qquad g = E_n\ \nabla_\Phi\ \ell_n(\Phi)$$

If with probability 1 over the draw of n we have $|(g_n)_i - g_i| \leq b$ for all i then with probability of at least $1-\delta$ over the draw of the sample

$$||\hat{g} - g|| \leq \frac{\eta}{\sqrt{k}} \qquad \eta = b\left(1 + \sqrt{2\ln(1/\delta)}\right)$$

Norkin and Wets "Law of Small Numbers as Concentration Inequalities ...", 2012, theorem 3.1

$$
\begin{aligned}
\ell(\Phi + \Delta\Phi) &\le \ell(\Phi) + g^\top \Delta\Phi + \frac{1}{2} L ||\Delta\Phi||^2 \\
\ell(\Phi - \eta\widehat{g}) &\le \ell(\Phi) - \eta g^\top \widehat{g} + \frac{1}{2} L\eta^2 ||\widehat{g}||^2 \\
&= \ell(\Phi) - \eta(\widehat{g} - (\widehat{g} - g))^\top \widehat{g} + \frac{1}{2} L\eta^2 ||\widehat{g}||^2 \\
&= \ell(\Phi) - \eta||\widehat{g}||^2 + \eta(\widehat{g} - g)^\top \widehat{g} + \frac{1}{2} L\eta^2 ||\widehat{g}||^2 \\
&\le \ell(\Phi) - \eta||\widehat{g}||^2 + \eta\frac{\eta}{\sqrt{k}} ||\widehat{g}|| + \frac{1}{2} L\eta^2 ||\widehat{g}||^2 \\
&= \ell(\Phi) - \eta||\widehat{g}|| \left(||\widehat{g}|| - \frac{\eta}{\sqrt{k}} \right) + \frac{1}{2} L\eta^2 ||\widehat{g}||^2
\end{aligned}
$$

Optimizing η

Optimizing η we get

$$||\widehat{g}|| \left(||\widehat{g}|| - \frac{\eta}{\sqrt{k}} \right) = -L\eta ||\widehat{g}||^2$$

$$\eta = \frac{1}{L} \left(1 - \frac{\eta}{||\widehat{g}||\sqrt{k}} \right)$$

Inserting this into the guarantee gives

$$\ell(\Phi - \eta\widehat{g}) \leq \ell(\Phi) - \frac{L}{2}\eta^2 ||\widehat{g}||^2$$

Optimizing k

Optimizing progress per sample, or maximizing η^2/k, we get.

$$\begin{aligned}
\frac{\eta^2}{k} &= \frac{1}{L^2}\left(\frac{1}{\sqrt{k}} - \frac{2\hat{\sigma}}{||\widehat{g}||k}\right)^2 \\
0 &= -\frac{1}{2}k^{-\frac{3}{2}} + \frac{2\hat{\sigma}}{||\widehat{g}||}k^{-2} \\
k &= \left(\frac{22\hat{\sigma}}{||\widehat{g}||}\right)^2 \\
\eta &= \frac{1}{2L}
\end{aligned}$$

END