TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

Stochastic Gradient Descent (SGD)

The Classical Convergence Thoerem

RMSProp, Momentum and Adam

Scaling Learning Rates with Batch Size

Gradient Flow and Langevin Dynamics

Vanilla SGD

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

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

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

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

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

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

The Classical Convergence Theorem

$$\Phi \mathbf{-=} \eta_t \nabla_\Phi \text{ loss}(\Phi, x_t, y_t)$$

For "sufficiently smooth" non-negative loss with

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

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.

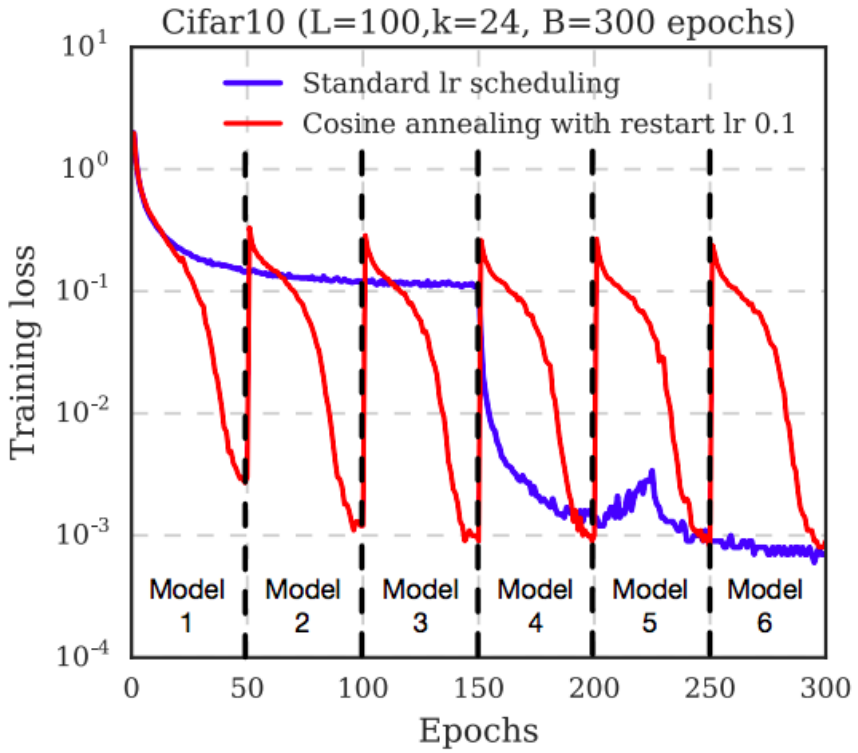
For sufficiently small learning rates any meaningful update of the parameters will be based on an arbitrarily large sample of gradients at essentially the same parameter value.

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 $x_1, x_2, x_3, \ldots$.

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

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

RMSProp

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

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

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

$$s[i] \approx E\, \hat{g}[i]^2 = g[i]^2 + \sigma[i]^2$$

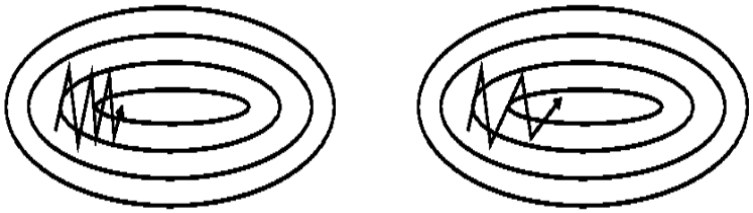
We should expect $\sigma[i] >> g[i]$.

Momentum

$$v_t = \left(1 - \frac{1}{N}\right) v_{t-1} + \eta * \hat{g}_t$$

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

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.

A Reparameterization

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

A Reparameterization

$$\mu_t = \left(1 - \frac{1}{N_\mu}\right)\mu_{t-1} + \frac{1}{N_\mu}\hat{g}_t \quad N_\mu \text{ typically 10 or 100}$$

$$\Phi_{t+1} = \Phi_t - N\eta\mu_t \tag{1}$$

$$= \Phi_t - \eta'\mu_t \tag{2}$$

My intuition: For the parameterization N and η' defined by (2) it should be possible to optimize N and η' largely independently. I would not expect this to be true for the N and η defined by (1).

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.

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

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

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 $x_1, \ldots, x_t$.

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

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

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.

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

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

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

Adam (simplified)

$$\mu_0[i] = s_0[i] = 0$$

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

$$s_t[i] = \left(1 - \frac{1}{\min(t, N_s)}\right) s_{t-1}[i] + \frac{1}{\min(t, N_s)} \hat{g}_t[i]^2$$

$$\Phi_{t+1}[i] = \Phi_t - \frac{\eta}{\sqrt{s_t[i] + \epsilon}} \ \mu_t[i]$$

Scaling η, N_μ and N_s 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

Scaling η

Consider two consecutive updates for a batch size of 1 with learning rate η_1.

$$\Phi_{t+1} = \Phi_t - \eta_1 \nabla_\Phi \text{loss}(\Phi_t, x_t, y_t)$$

$$\Phi_{t+2} = \Phi_{t+1} - \eta_1 \nabla_\Phi \text{loss}(\Phi_{t+1}, x_{t+1}, y_{t+1})$$

$$\approx \Phi_{t+1} - \eta_1 \nabla_\Phi \text{loss}(\Phi_t, x_{t+1}, y_{t+1})$$

$$= \Phi_t - \eta_1((\nabla_\Phi \text{loss}(\Phi_t, x_t, y_t)) + (\nabla_\Phi \text{loss}(\Phi_t, x_{t+1}, y_{t+1})))$$

Scaling η

Let η_B be the learning rate for batch size B.

$$\Phi_{t+2} \approx \Phi_t - \eta_1((\nabla_\Phi \text{loss}(\Phi_t, x_t, y_t)) + (\nabla_\Phi \text{loss}(\Phi_t, x_{t+1}, y_{t+1})))$$

$$= \Phi_t - 2\eta_1 \, \hat{g} \quad \text{for} \quad B = 2$$

Hence two updates with $B = 1$ at learning rate η_1 is the same as one update at $B = 2$ and learning rate $2\eta_1$.

$$\eta_2 = 2\eta_1, \qquad \eta_B = B\eta_1$$

Scaling N_μ

Let $N_{\mu,B}$ be the momentum parameter to be used with batch size B.

For batch size B, $\hat{\mu}_t$ is averaging over $N_{\mu,B}B$ gradient values.

Holding the number of included gradients constant gives

$$N_{\mu,B}B = N_{\mu,1} \quad \text{or} \quad N_{\mu,B} = N_{\mu,1}/B$$

Scaling N_s

The simple analysis for N_μ fails for N_s.

To estimate $E\ g[i]^2$ we should average $\hat{g}_t[i]^2$ over batch elements rather than batch averages.

The parameter N_s should be a number of gradients (batch elements) rather than a number of batches,

Under this semantics, N_s should be constant independent of batch size.

Gradient Flow

Consider the differential equation

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

Let $\Phi(t)$ be the solution satisfying $\Phi(0) = \Phi_{\text{init}}$.

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

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

For a given value of t and N let Φ_j be defined by

$$\Phi_0 = \Phi_{\text{init}}$$

$$\Phi_{j+1} = \Phi_j - \left(\frac{t}{N}\right) g(\Phi_j)$$

$$\lim_{N\to\infty} \Phi_N = \Phi(t)$$

Progress Theorem for Gradient Flow

$$\begin{aligned}
\frac{d\ell}{dt} &= (\nabla_\Phi \, \ell(\Phi)) \cdot \frac{d\Phi}{dt} \\
&= -(\nabla_\Phi \, \ell(\Phi)) \cdot (\nabla_\Phi \, \ell(\Phi)) \\
&= -||\nabla_\Phi \, \ell(\Phi)||^2 \\
&\leq 0
\end{aligned}$$

If $\ell(\Phi) \geq 0$ then $\ell(\Phi)$ must converge to a limiting value.

This does not imply that Φ converges.

SGD also yields Gradient Flow

For

$$\Phi_{j+1} = \Phi_j - \left(\frac{t}{N}\right) \hat{g}_j$$

we still have

$$\lim_{N \to \infty} \Phi_N = \Phi(t)$$

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.

Langevin Dynamics

Consider actual gradient descent

$$\Phi_{j+1} = \Phi_j - \eta \hat{g}_j$$

Inspired by gradient flow, we define time by $t = \eta i$.

For Δt large compared to η (so that it corresponds to many SGD updates), but still small enough so that the gradient does not change as the updates are made, we have that $\Phi(t + \Delta t)$ is distributed as

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

where Σ is the covariance matrix of the random vector $\hat{g}$.

Langevin Dynamics

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

$$\Sigma = E_i \left(\hat{g}_i - g\right)\left(\hat{g}_i - g\right)^\top$$

It turns out that one can define a continuous time stochastic process — Langevin dynamics — defined by the notation

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

This is a stochastic differential equation.

If $\mathcal{L}(\Phi)$ is constant (a plateau) and $\Sigma(\Phi)$ is constant we get Brownian motion.

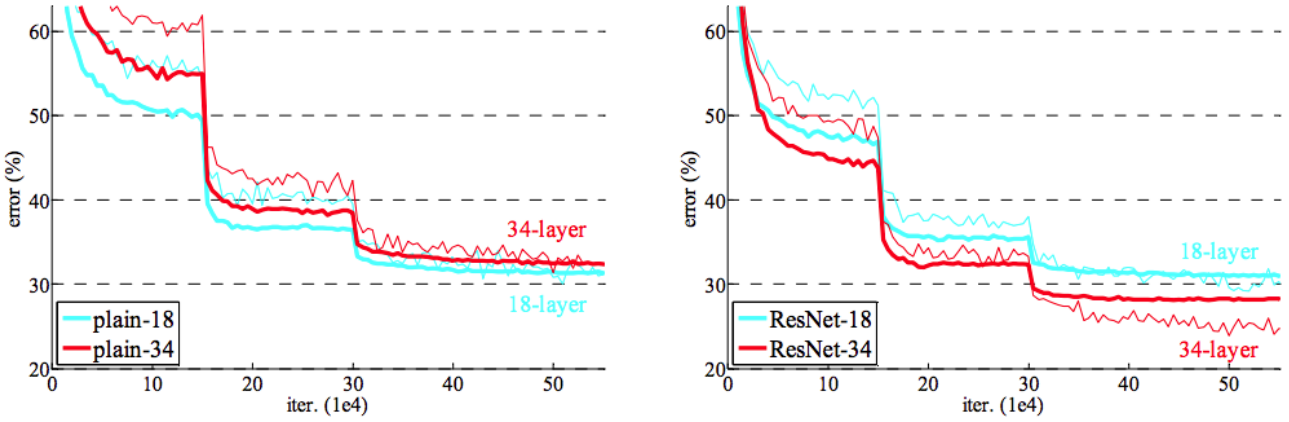
Langevin Dynamics (Without Annealing)

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

$$\Sigma = E_i \left(\hat{g}_i - g\right)\left(\hat{g}_i - g\right)^\top$$

Note that as the learning rate $\eta \to 0$ the noise term vanishes and we are back to gradient flow.

Annealing the Learning Rate



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.

Stationary Distributions

SGD defines a Markov process in parameter space.

$$\Phi_{i+1} = \Phi_i - \eta \hat{g}_i$$

Under Langevin dynamics the stationary distribution of the Markov process must satisfy a certain differential equation which can be solved. This gives a general form for Langevin stationary distributions.

In one dimension the stationary distribution is a Boltzman distribution. But this is not generally true in dimension higher than 1.

Langevin Dynamics With Annealing

Although the stationary distribution is not a Boltzman distribution, the learning rate η acts as a temperature parameter in the sense of the last line below.

Let P_η be the stationary distribution at learning rate η.

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

$$\mathcal{L}(\eta) = E_{\Phi \sim P_\eta} \, \mathcal{L}(\Phi)$$

$$\lim_{\eta \to 0} \mathcal{L}(\eta) = \mathcal{L}(\Phi^*) \quad \text{for Langevin dynamics}$$

A Quadratic Basin Theorem

In practice annealed SGD will converge to a local optimum.

We consider stationary distributions restricted to a neighborhood of a (local) optimim Φ^* satisfying

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

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

$$E_i \, g_i = 0$$

$$E_i \, H_i \qquad \text{is positive definite}$$

A Quadratic Basin Theorem

$$
\begin{aligned}
\mathcal{L}(\Phi) &= E_i\ \mathcal{L}_i(\Phi) \\
\mathcal{L}(\eta) &= E_{\Phi \sim P_\eta}\ \mathcal{L}(\Phi) \\
\mathcal{L}_i(\Phi^* + \Delta\Phi) &= \mathcal{L}_i + g_i \Delta\Phi + \frac{1}{2} \Delta\Phi^\top H_i \Delta\Phi \\
E_i\ g_i &= 0 \\
E_i\ H_i & \quad \text{is positive definite}
\end{aligned}
$$

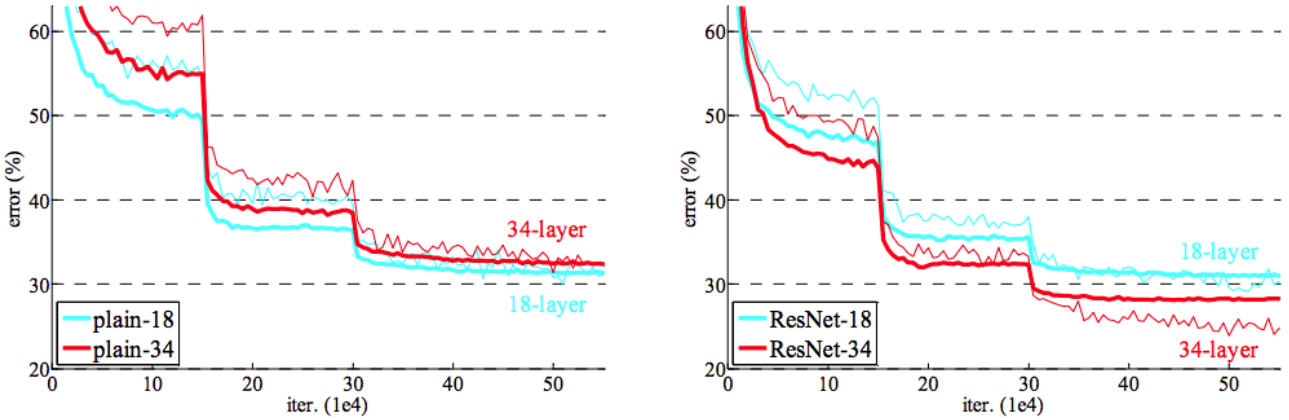
Theorem:

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

Analyzing Stationary Distributions

$$\mathcal{L}(\eta) \approx \mathcal{L}(\Phi^*) + \eta \left(E_I ||g_i||^2 \right)$$

$$\mathcal{L}(\Phi^*) = \mathcal{L}(\eta) - 2(\mathcal{L}(\eta) - \mathcal{L}(\eta/2)) = \mathcal{L}(\eta) - 2\mathcal{L}(\eta/2)$$



Proof: Observation 1

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

Proof: Step 1

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 2

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

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

$$= \frac{1}{4} E_i \lim_{\eta \to 0} E_{\Delta\Phi \sim P_\eta} ||(g_i + H_i \Delta\Phi)||^2$$

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

END