POLITECNICO DI TORINO

Master degree course in Physics of Complex Systems

Master Degree Thesis

Optimization through control based gradient descent

Supervisors
prof. Hilbert Johan Kappen
prof. Alfredo Braunstein

Candidate
Paolo Pavanelli
matricola s234131

Accademic Year 2017-2018

Summary

Non-convex continuous optimization problems occur in many fields of engineering, ranging from operations research, control theory and neural network learning. The most common optimization approach are gradient based method that,however, for large optimization problems, these approaches are plagued by local minima and saddle points,that is suboptimal solutions where the gradients are zero and where optimization halts prematurely. Recently, Baldassi et al. [1] have introduced a *local entropic measure* for learning with discrete synapses that leads to unanticipated computational performance.
Inspired by this line of work, Chaudhari [2] connects this idea to the solution of a Hamilton-Jacobi partial differential equation and stochastic optimal control theory [4], providing an algorithm to implement the descent along the gradient of the local entropy.
Here we provide an algorithm based on the work on learning parametrized controllers done by Kappen and to compare its performances to the one proposed by Chaudhari [3].
We explore the validity of our method in relevant one dimensional cases and in multidimensional cases of both a convex and a non convex function, finding that in both cases the control descent gives a comparable extimate of the global minimum of the original function given the same fixed parameters and thus suggesting that this method could prove itself as a valid alternative with further reasearch.

Contents

List of Figures

Chapter 1

Introduction

Non-convex continuous optimization problems occur in many fields of engineering. Examples are in operations research, control theory and neural network learning. The most common optimization approach are gradient based method. For large optimization problems, these approaches are plagued by local minima and saddle points, suboptimal solutions where the gradients are zero and where optimization halts prematurely.

1.1 State of the art

In literature it is possible to find different methods to perform global optimization based on physics, among which we can recall the simulated annealing [6] and parallel tempering [7]. Recently, Baldassi et al. [1] have introduced a *local entropic measure* that can be used as an alternative of global optimization.
Inspired by this line of work, Chaudhari [2] connects this idea to the solution of a Hamilton-Jacobi partial differential equation and stochastic optimal control theory [4], providing an algorithm to implement the descent along the gradient of the local entropy.
In their work they both explore the possibility of finding not the global minimum but a wide valley of local minima, which leads to better results in implementing learning algoritms.

However their suggested procedure can easily be tuned in to perform a global optimization problem since it simply depends on the choice of some parameters. To understand this better we will now proceed in a brief summary of the procedure explained in [2].
Consider a general optimization problem of the form

$$w^* = \text{argmin}_w f(w) \tag{1.1}$$

where $w \in \mathbb{R}^n$ and f is a non-convex cost objective. Minimizing f is usually done using a gradient based method from a random initial value: $w_{t+1} = w_t - \eta \nabla f(w)$ with $\nabla f(w)$ the gradient of f at w.
Instead of minimizing f,minimize a local entropy distribution of the form:

$$F(w, \gamma, \beta) = -\log \int dx p(x|w) e^{-\beta f(x)} \qquad p(x|w) \rightarrow e^{-\beta \frac{\gamma}{2} \|x-w\|^2} \tag{1.2}$$

where $\| \cdot \|^2$ denotes Euclidean norm.
This form is derived from the interpretation of $f(w)$ as an energy landscape, which leads to a formulation of the Gibbs distribution in which β takes the role of an inverse temperature and γ is a parameter that is linked with the convexity of $F(w)$.
Infact for $\gamma \rightarrow 0$ the local entropy will be almost convex, instead for $\gamma \rightarrow \infty$, $F(w, \beta, \gamma)$ will result of the same shape of $f(w)$.
In [2] β is assumed equal to 1 since the goal is not to reach the global minimum of $f(w)$ but to reach a wide valley of local minima; for our purposes instead we will keep the parameter since for $\beta \rightarrow \infty$ the Gibbs distribution concetrates above the global minimum of $f(w)$, thus preserving the global minimum also in $F(w, \gamma)$.
However this minimization presents a problem in the sense that the analitycal computation of $F(w)$ is not always straight-forward, so in order to perform a gredient descent along the local entropy it is necessary to estimate the value of $\nabla F(w)$.
In [2] this estimate is expressed in the formula:

$$-\nabla F(w, \gamma, \beta) = \gamma(w - \langle x \rangle) \tag{1.3}$$

Where $\langle \bullet \rangle$ denotes the expectation value over the Gibbs distribution of the original function $f(w)$. This expectaction value is actually estimated through a Langevin dynamics, generated by the following equation:

$$dx' = \nabla f(x') - \gamma(x' - w) + dw_t \tag{1.4}$$

with $\langle dw_t^2 \rangle = \frac{1}{2\beta} dt$. Then the $\langle x \rangle$ in 1.3 can be approximated to $\frac{1}{N} \sum_{t=1}^{N} x'_t$.

Thanks to this estimate it is now possible to perform a gradient descent on $F(w)$,by taking into account that it is only a local estimate for the gradient so it is necessary a double loop: The outer loop, which actually performs the gradient descent on $F(w)$ and gives the value of the desired w^*; the inner loop, which estimates locally the gradient through the langevin dynamics at each update.

1.2 Project Layout

In this project is proposed to interpret $p(x|w) = p(X_T = x|X_0 = w)$ as the conditional distribution *after a finite time* T of a Brownian motion

$$dX_t = dW_t \qquad X_0 = w \tag{1.5}$$

with dW_t Gaussian white noise with mean zero $\mathbb{E}dW_t = 0$ and unit variance $\mathbb{E}[dW_t^2] = dt$.
$\mathbb{E}$ denotes expectation value.
With these assumptions we can relate Eq. 1.2 to a finite horizon stochastic optimal control problem on the time interval $t \in [0, T]$ as will be explained in the following section. This link allow us to estimate the gradient of $F(w)$ in a different way than the one proposed by Chaudhari in [2], which consists of finding the optimal control of the above stated problem. In fig. 1.1 we can appreciate

for a random one dimensional function $f(w)$ the different landscapes that $F(w)$ assumes by fixing the time horizon T, which corresponds to fixing γ in the Chaudhari formulation. We plot $F(w)$ for $T = 0.01$ and $T = 0.1$. The free energy smoothes the local minima and maxima while the global minimum is kept the same as $f(w)$. Note, that varying T sets the spatial scale at which the local minima are removed, but to preserve the same global minimum as $f(w)$ it is necessary to fix β large enough.

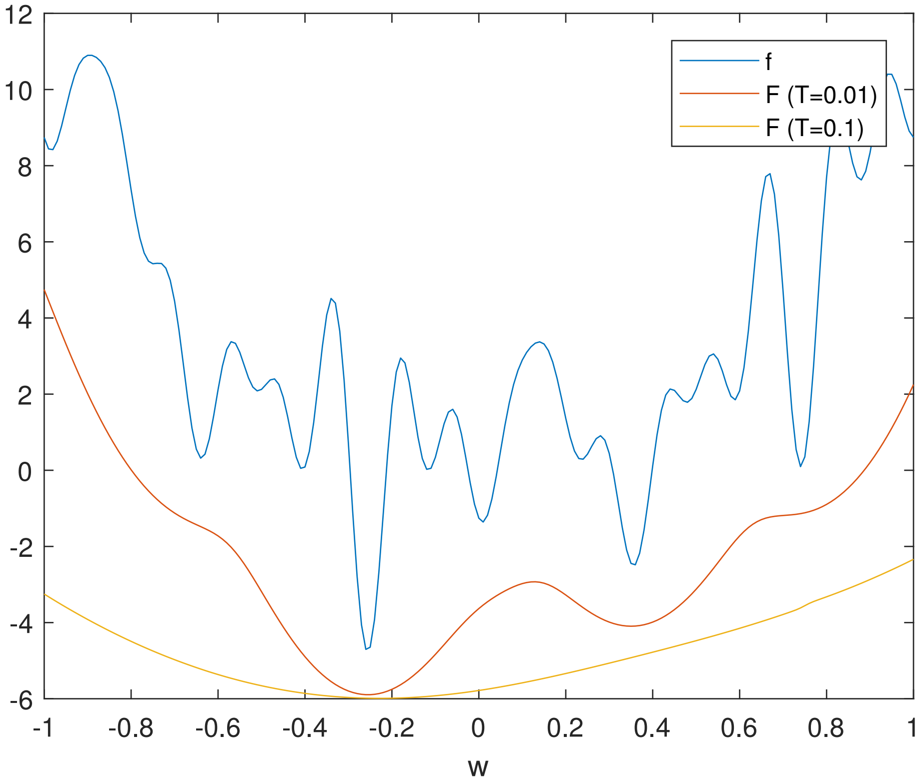

Figure 1.1. Left: The one dimensional function $f(w)$ has many local minima on the interval $[-1, 1]$. The free energy expression Eq. 1.2 for $T = 0.01$ and $T = 0.1$ has less local minima. Note, that nevertheless the global minima coincide.

1.3 Path integral control

In this section, are reviewed some well-known facts about the path integral control method. See [3] for more details and references to the literature.

Consider the following dynamical system on the time interval $t \leq s \leq T$ and cost S_u of a trajectory $\tau = x_{[t:T]}$:

$$dX_s = h(X_s, s)ds + \sigma(X_s, s)\left(u(X_s, s)ds + dW_s\right) \qquad X_t = x \tag{1.6}$$

$$S_u(\tau|x,t) = f(X_T) + \int_t^T dsV(X_s, s) + \int_t^T ds\frac{1}{2}\|u(X_s, s)\|^2 + \int_t^T dsu(X_s, s)dW_s \tag{1.7}$$

The stochastic optimal control problem is to find the optimal control function u:

$$\begin{aligned} J(x,t) &= \min_u \mathbb{E}_u \; S_u(\tau|x,t) \\ u^*(x,t) &= \arg\min_u \mathbb{E}_u \; S_u(\tau|x,t) \end{aligned} \tag{1.8}$$

that minimizes the expected control cost $C(x,t) = \mathbb{E}_u S_u$, where $\mathbb{E}_u$ is an expectation value with respect to the stochastic process Eq. 1.6 with control u.

$J(t,x)$ is called the optimal cost-to-go and is the optimal cost from any intermediate state x and any intermediate time t to the end time T. For any control problem, J satisfies a partial differential equation known as the Hamilton-Jacobi-Bellman equation (HJB). In general, this control problem is very hard to solve. The above control problem is a so-called path integral control problem [4, 5], whose optimal solution is given in terms of a path integral

$$J(x,t) = -\log\psi(x,t) \qquad \psi(x,t) = \mathbb{E}_u \; e^{-S_u(\tau|x,t)} \tag{1.9}$$

The optimal control is given as $u(x,t) = -\sigma(x,t)^T \nabla J(x,t)$.

Note, that the optimal cost-to-go involve an expectation over a stochastic process with a control function u. It states that these expectations are independent of the value of the control u. All controls give the same unbiased estimate of ψ. However, their variance differ. It can be shown that the closer u is to optimal control the smaller the variance. When $u = u^*$, the optimal control, the sampling procedure is optimal in the sense that the variance of the estimator is zero.

Denote $p_u(\tau|x,t)$ the probability density of trajectories τ under the dynamics Eq. 1.6 with control u. Then from Eq. 1.9

$$\begin{aligned} J(x,t) = & -\log \sum_\tau p_0(\tau|x,t) e^{-S_0(\tau|x,t)} = -\log \sum_\tau p_u(\tau|x,t) e^{-S_0(\tau|x,t) - \log \frac{p_u(\tau|x,t)}{p_0(\tau|x,t)}} \\ \leq & \sum_\tau p_u(\tau|x,t) \left(S_0(\tau|x,t) + \log \frac{p_u(\tau|x,t)}{p_0(\tau|x,t)} \right) = \sum_\tau p_u(\tau|x,t) S_u(\tau|x,t) \end{aligned} \tag{1.10}$$

where we used Jensens' inequality and the fact that the last expression between brackets is equal to the expression in Eq. 1.7. Eq. 1.10 simply states that the optimal-cost-to-go $J(x,t)$ is less than the expect cost $C(x,t)$ using any sub-optimal control.

The inequality Eq. 1.10 is saturated when $S_u(\tau|x,t)$ has zero variance, in which case $J(x,t) = C(x,t) = S_u(\tau|x,t)$. The value of $u = u^*$ for which this occurs is the optimal control[1]. Note from Eq. 1.9 that $\psi(x,t)$ is the normalisation constant of the distribution

$$p^*(\tau|x,t) = \frac{1}{\psi(x,t)} p_u(\tau|x,t) e^{-S_u(\tau|x,t)} \tag{1.11}$$

which is the distribution over trajectories under the optimal control [2].

[1] The mathematical condition for this control to exist is that $\sum_\tau p_0(\tau|, xt) e^{-S_0(\tau|x,t))} < \infty$.

[2] From the above arguments it follows that p^* is independent of u. When $u = u^*$, $\psi = e^{-S_u(\tau|x,t)}$ and $p^*(\tau|x,t) = p_u(\tau|x,t)$.

Chapter 2

Control Based Gradient Descent

With the above control formulation, it is possible now identify the free energy F in Eq. 1.2 as the optimal cost-to-go in Eq. 1.9.
Consider a controlled Brownian motion on the interval $t \in [0, T]$ with end cost:

$$dX_t = u(X_t, t)dt + dW_t \qquad\qquad X_0 = w \tag{2.1}$$

$$S_u(\tau|w) = f(X_T) + \int_0^T dt \frac{1}{2}\|u(X_t, t)\|^2 + \int_0^T dt u(X_t, t)dW_t \tag{2.2}$$

Since the path cost $V = 0$, the (uncontrolled) marginal distribution at $t = T$ is Gaussian: $p_0(x_T, T|w{,}0) = \mathcal{N}(x_T|w, T)$ and $S_0(\tau|w) = f(x_T)$.
The optimal cost-to-go $J(w{,}0)$ Eq. 1.9 is equal to the free energy $F(w)$ in Eq. 1.2 with $\sigma^2 = T$. The optimal control is

$$u(w) = -\nabla J(w{,}0) = \frac{\langle X_T \rangle - w}{T} \qquad\qquad \langle X_T \rangle = \int d\tau x p^*(x|w) \tag{2.3}$$

$\langle X_T \rangle$ is the expected value of the X_T under p^*. Note, that $\langle X_T \rangle$ depends on w. Otherwise, the control solution would steer in a straight line from w to $\langle X_T \rangle$ in time T.
When $T \to 0$, $F(w) \to f(w)$ and we recover the original gradient

descent algorithm. The finite horizon control picture is illustrated in fig. 2.1, where we plot F as a function of w and horizon time T.

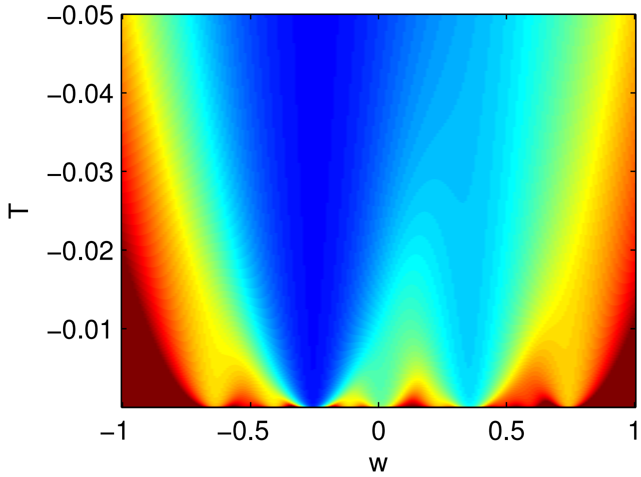

Figure 2.1. The free energy F is the optimal cost-to-go for a finite horizon problem with horizon time T and end cost $f(w)$. The figure plots F as a function of w and horizon time T.

The figure is somewhat misleading. Although it is true that F is globally minimized and F is much smoother than f, the gradient is computed only locally around the current w. This is illustrated for the same function in fig. 2.2. On the left we plot the local energy $E(x|w) = \log p_0(x|w)e^{-f(x)}$ for fixed $w = -0.8$ versus x for different values of $T = 0.001$,0.01,0.1. We see that for short horizon T the gradient steers towards the nearest local minimum around $x = -0.65$ and for larger T towards the global minimum.

This effect can be better seen in fig. 2.2 right. The minimization of f is not affected by a global scaling β, but does effect the free energy. In particular in order to preserve the same global minimum β should be large enough. We replace T/β with T. The the free energy becomes:

$$F(w) = -\log \int dx e^{-\beta f(x)} e^{-\|x-w\|^2/2T} \tag{2.4}$$

By plotting $\langle X_T \rangle$ versus T for different values of β we see that β

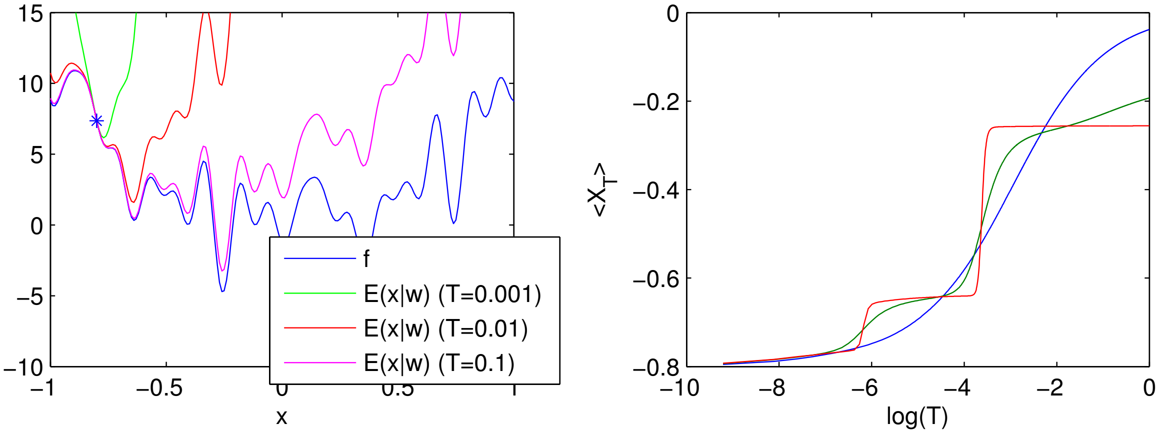

Figure 2.2. Left: The local energy $E(x|w) = \log p(x|w)e^{-f(x)}$ versus x for $w = -0.8$ and different horizon times $T = 0.001$ (blue), $T = 0.01$ (green) and $T = 0.1$ (magenta). Right: $\langle X_T \rangle$ versus T for different values of β.

should be sufficiently large, which is in agreement with what we have stated before.

As Eq. 2.3 and Eq. 1.11 indicate, we can estimate $\langle X_T \rangle$ by sampling from Eq. 1.5 and weighting each trajectory with $e^{-\beta f(X_T)}$. We can improve the sampling efficiency by sampling from the controlled dynamics Eq. 2.1. In [3] we propose to use the Cross Entropy method to learn an arbitrary parametrized controller, yielding an adaptive importance sampling scheme.

2.1 Constant Control

Here, we propose the simplest possibility, with u constant independent of x, t, so we only have to estimate the vector u itself. In this case we can integrate Eq. 2.1 which yields $X_T = w + uT + W_T$, with W_T a mean zero Gaussian with variance T. The optimal distribution and approximating distributions are

$$p^*(x|w) = \frac{1}{\psi(w)} e^{-\frac{\|x-w\|^2}{2T} - f(x)} \qquad p_u(x|w) = \frac{1}{\sqrt{2\pi T}} e^{-\frac{\|x-w-uT\|^2}{2T}} \tag{2.5}$$

We find u by minimizing the Cross Entropy criterion [3]

$$KL(p^*|p_u) = \int dx p^*(x|w) \log \frac{p^*(x|w)}{p_u(x|w)} \propto \mathbb{E}_{\hat{u}} e^{-S_{\hat{u}}} \|X_T - w - uT\|^2 \tag{2.6}$$

$\hat{u}$ is an importance sampling control that can have any value. By seting $\frac{\partial KL(p^*|p_u)}{\partial u_i} = 0$ we estimate u as

$$u = \hat{u} + \frac{1}{T} \frac{\mathbb{E}_{\hat{u}} e^{-S_{\hat{u}}} W_T}{\mathbb{E}_{\hat{u}} e^{-S_{\hat{u}}}} \qquad S_{\hat{u}} = f(X_T) + \frac{1}{2} T \|\hat{u}\|^2 + \hat{u} W_T \tag{2.7}$$

When $\hat{u} = u^*$ is the optimal control $S_{\hat{u}}$ has zero variance so that $\mathbb{E}_{\hat{u}} e^{-S_{\hat{u}}} W_T = e^{-S_{\hat{u}}} \mathbb{E}_{\hat{u}} W_T = 0$ and Eq. 2.7 becomes $u = u^*$, ie. the CE procedure estimates the optimal control. We choose $\hat{u} = u$ the current estimated value of the optimal control.

We can construct two algorithms from the above ideas. The first is to do treat F as the new cost objective and do gradient descent in F, where the gradient is given by u in Eq. 2.7. This yields the following iterative algorithm

- Initialize w. Initialize $u = 0$. Choose T, remembering that this value should be large enough to smooth the local minima, nut not too large that the control problem is no longer considerable to be at finite horizon [from our simulations a good value is $T \in [0.1,1]$]. Choose β large enough [in the following simulations $\beta = 5$, as we seen from 2.2]

- In each iteration,
  - draw M samples W^m from a mean zero Gaussian with circular variance T.
  - Compute $x^m = w + uT + W^m$ and $S^m = f(x^m) + \frac{1}{2}T\|u\|^2 + uW^m$. Estimate the partition sum as $\hat{\psi} = \frac{1}{M}\sum_{m=1}^{M} e^{-S^m}$.
  - Update the control

$$u := u + \frac{1}{T}\frac{\frac{1}{M}\sum_{m=1}^{M} W^m e^{-S^m}}{\hat{\psi}} \tag{2.8}$$

    This provides the gradient because $u = -\frac{\partial F}{\partial w}$.
  - Do the gradient step: $w_{i+1} = w_i + \eta u T$. (We could do $w := w + \eta u$ but find it convenient to rescale η with T.)
  - Since $uT = \langle X_T \rangle - w$ and we have adapted w we should adapt $u := (1 - \eta)u$.

The method is applied to the one dimensional function of fig. 1.1, with different initializations. We use 15 iterations and $m = 500$ samples per iteration. The results for different T are given in fig. 2.3.

As we can see in the fig.. 2.3 for small values of T some of the trajectories still end up in some local minima instead of steering towards the global minimum, so in order to minimize the original $f(w)$ one logical step to follow would be to use a large value of T.

After setting this value it is true that the trajectories are converging to the global minima, but as one can appreciate also from fig.. 2.4, what we are minimizing with the previous algorithm is actually $F(w)$ and not $f(w)$. Since β is large enough the global minima coincide and our algorithm finds a minimum of $w = -0.2558$, which is in agreement with the value of$w^* = -0.26$ found for $f(w)$.

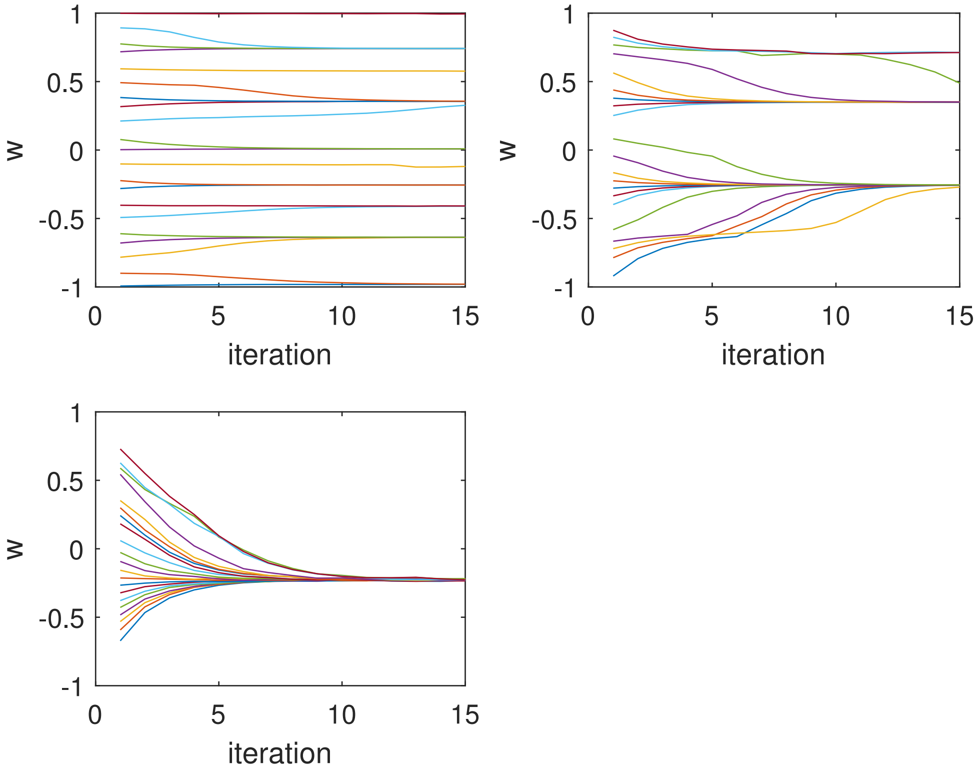

Figure 2.3. Top: on the left is represented the descent done with the algorithm for T=0.001, on the right T=0.01; Bottom: Descent with T=0.1

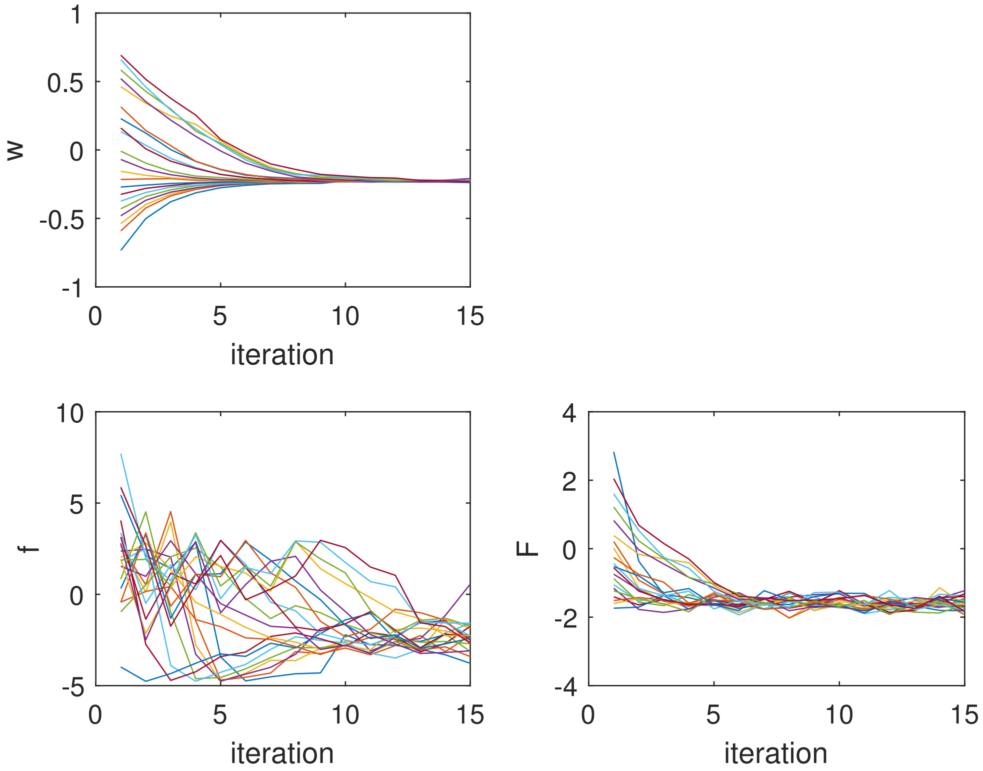

Figure 2.4. Top: Descent with T=0.1; Bottom: On the left it is shown tha value of f for each iteration, on the right the value of F for each iteration

2.2 Multidimensional

Now that we have tuned in a procedure that is working for one dimensional function, we would like to see if such algorithm performs well even in higher dimensions.
In order to do that we first want to verify its validity in the simplest possible case which is the convex one and then see if for a non convex $f(w)$ the results are comparable to the one found using the Langevin extimate of $\nabla F(w)$ (LBGD).

2.2.1 Convex

Be n the number of dimensions. Consider the function $f(w) = \sum_{i=1}^{n} \frac{1}{2}\alpha_i w_i^2$ where w_i are the components of the n-dimensional vector w. In this case it is possible to compute analitically the value of $F(w)$.
Once done that we can apply the algorithm tested in one dimension and check that the gradient descent on $F(w)$ and the one done through the control have the same trajectory. Then we can compare them to the estimate done trhough Langevin dinamycs.

$$F(w) = -\log \int dx e^{-\frac{||x-w||^2}{2T}} e^{-\beta f(x)} = \tag{2.9}$$

$$= -\log \int dx e^{-\beta \sum_{i=1}^{n} \frac{\alpha_i x_i^2}{2}} e^{-\frac{||x-w||^2}{2T}} = \tag{2.10}$$

$$= -\log \int dx e^{-\beta \frac{\sum_{i=1}^{n} \alpha_i x_i^2}{2}} e^{-\frac{\sum_{i=1}^{n} (x_i - w_i)^2}{2T}} = \tag{2.11}$$

$$= -\log e^{-\frac{\sum_{i=1}^{n} w_i^2}{2T}} \int dx e^{-\frac{\sum_{i=1}^{n} x_i^2 (\beta\alpha_i + \frac{1}{T})}{2}} e^{\frac{\sum_{i=1}^{n} 2x_i w_i}{2T}} \tag{2.12}$$

Let's now introduce: the matrix A$\in \mathbb{R}^{m\times m}$ defined as $A_{ij} = 0 \quad \forall i \neq j$ and $A_{ii} = (\frac{\beta\alpha_i T+1}{T})$; the vector B $\in \mathbb{R}^m$ defined as

$B_i = \frac{w_i}{T}$ then:

$$F(w) = -\log e^{-\frac{||w||^2}{2T}} \int dx e^{-\frac{1}{2}x^T A x} e^{\frac{1}{2}B^T x} = \tag{2.13}$$

$$= -\log e^{-\frac{||w||^2}{2T}} e^{\frac{1}{2}B^T A^{-1} B} \sqrt{\frac{2\pi^m}{det(A)}} = \tag{2.14}$$

$$= \frac{1}{2T} \sum_{i=1}^{m} w_i^2 (\frac{\beta\alpha_i T}{\beta\alpha_i T + 1}) + log(\sqrt{\frac{2\pi^m}{det(A)}}) \tag{2.15}$$

From 2.13 to 2.14 we used tha formula for gaussian integrals and then computed the result for our matrix A and our vector B.
With the explicit form of $F(w)$ it is possible now to compute its gradient components, which are : $\frac{\partial F}{\partial w_i} = w_i(\frac{\beta\alpha_i}{\beta\alpha_i T+1})$

We proceed then to apply our previous algorithm to this $f(w)$ with $n = 2$.
In particular we chose the fucntion of this form $f(w) = w_1^2 + 25w_2^2$. We fix $T = 0.5$ and $\beta = 5$. The global minimum of $F(w)$ and $f(w)$ are the same: $w^* = [0,0]$. This simulation is performed by setting the number of iteration to 150 and $m = 500$.

We start all trajectories in $w = [3,3]$ but then we follow four different updates for each trajectory. One trajectory, the blue one in fig.2.5 follows $w^{r+1} = w^r + \eta u$ which corresponds to control based gradient descent (CBGD), the second one (orange in fig.2.5) follows $w^{r+1} = w^r - \nabla F(w)$ a gradient descent along $F(w)$, the third one (yellow in fig.2.5) $w^{r+1} = w^r - \nabla f(w)$ a gradient descent along $f(w)$ and the last one (purple in fig.2.5) which follows the algorithm proposed by Chaudhari .
In this simulation we set $\eta = 0.03$; it is needed such a small value otherwise the taylor expansion needed for the gradient descent along $f(w)$ wouldn't be true and then we couldn't make the comparison.

The results obtained are $w = [-0.005,0.001]$ for CBGD, $w = [0.004,0.001]$ for GD along $F(w)$, $w = [-0.001,-0.001]$ for the

LBGD and $w = [0.03{,}0]$ for the GD on $f(w)$.

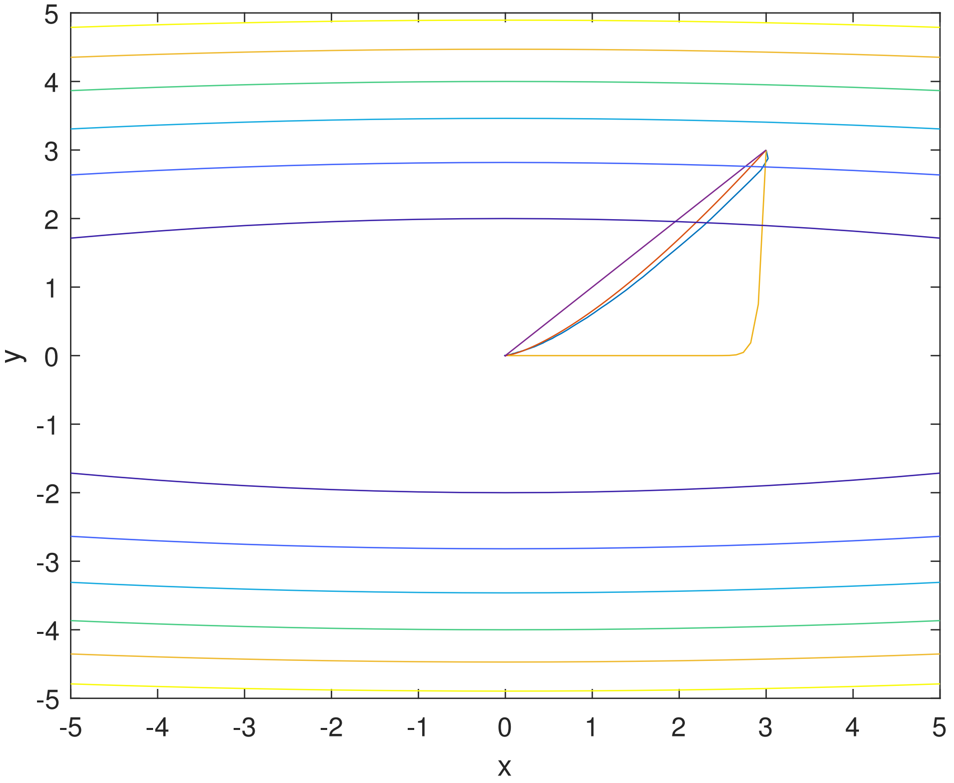

Figure 2.5. Contour plot of the path done by gradient descent on $f(w)$ in yellow, on $F(w)$ in orange,SLGD on $F(w)$ in black and with control descent in blue

From the results stated above we can see how all the gradient descent performed along $F(w)$ give comparable results, while the descent on $f(w)$ gives a result which is 10 times bigger.
From fig.2.5 we can also see how in the long run the CBGD actually gives a better extimate on $\nabla F(w)$ than SBGD. This is actually dued to the fact that at each step the control updates itself, thus giving a better estimate of $\nabla F(w)$ while approaching the optimal control.

2.2.2 Non-Convex

Once we checked that our code is indeed working on the convex case and is giving comparable results to the LBGD we can carry out a comparison for a non-convex $f(w)$.

For this simulation we consider $dim = 2$ and

$$f(w) = \sqrt{(x - sin(2x+3y) - cos(3x-5y))^2 + (x - sin(x-2y) + cos(x+3y))^2}$$

with $w = [x, y]$

In order to compare CBGD, LBGD algorithms to the regular gradient descent we compute analitically the gradient of this function and then we let the trajectories evolve from the same starting point, each one of them with its update rules.

$$\frac{\partial f}{\partial x} = \frac{(x - sin(x-2y) + cos(x+3y))(-sin(x+3y) - cos(x-2y) + 1)}{\sqrt{(x - sin(2x+3y) - cos(3x-5y))^2 + (x - sin(x-2y) + cos(x+3y))^2}} + \tag{2.16}$$

$$+ \frac{(3sin(3x-5y) - 2cos(2x+3y) + 1)(-sin(2x+3y) - cos(3x-5y) + x)}{\sqrt{(x - sin(2x+3y) - cos(3x-5y))^2 + (x - sin(x-2y) + cos(x+3y))^2}} \tag{2.17}$$

$$\frac{\partial f}{\partial y} = \frac{(-sin(x-2y) + cos(x+3y) - x)(2cos(x-2y) - 3sin(x+3y))}{\sqrt{(x - sin(2x+3y) - cos(3x-5y))^2 + (x - sin(x-2y) + cos(x+3y))^2}} + \tag{2.18}$$

$$+ \frac{(-5sin(3x-5y) - 3cos(2x+3y))(-sin(2x+3y) - cos(3x-5y) + x)}{\sqrt{(x - sin(2x+3y) - cos(3x-5y))^2 + (x - sin(x-2y) + cos(x+3y))^2}} \tag{2.19}$$

Now that we have computed the gradient we can start our updates; for this simulation we set $m = 500$, and $\eta = 0.05$. We fix $T = 0.8$ and $\beta = 5$, still have a value large enough in order to make the global minima coincide.

Starting the trajectories from $w = [1,1]$ after 100 iterations we obtain $w = [0.007,1.28]$ with control based gradient descent, $w = [0.07,1.38]$ with Langevin estimate of $\nabla F(w)$ and $w = [1.55,1.18]$ with gradient descent along $f(w)$; the global minimum found using matlab min function is $w = [0.1,1.1]$

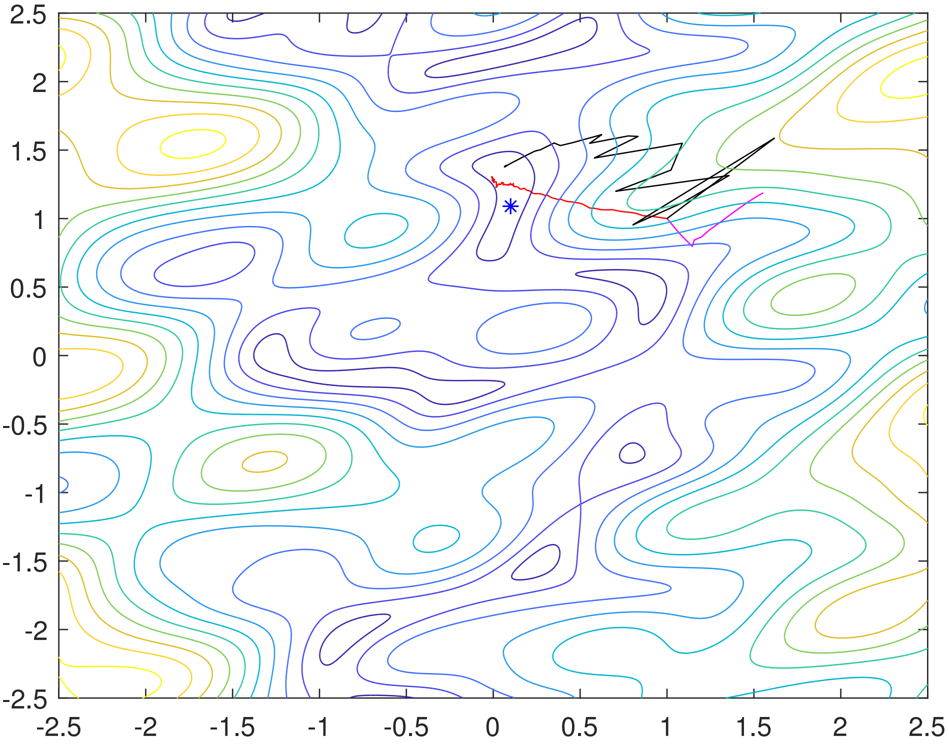

Figure 2.6. Contour plot of the path done by gradient descent on $f(w)$ in magenta, with control based descent in red and LBGD in black; the blue dot represents the global minimum

Looking at the fig.2.6 we have the confirmation that both algorithms performs considerably better than the gradient descent along $f(w)$ since the trajectories actually go towards the global minimum instead of getting stuck in some local minima.
The results obtained through LBGD and CBGD are also comparable, and in order to have a numerical result to support this statement, we compute the distance $\|w^*_{algorithm} - w^*_{global}\|$ where with w^*_{global} we intend [0.1,1.1].
The distances comuputed in this way are 0.21 for the CBGD and 0.29 for LBGD.
This result confirms what one could think by seeing the fig. 2.6, that is both trajectories end up in a neighborhood of the global minimum.

Conclusions

From the previous sections we can see that our method gives results that are comparable to the one proposed by Chaudarhi in finding a global minimum of an arbitrary function $f(w)$ for the same fixed parameters.

This gives us certainty on the validity of our algorithm, and opens up different possibilities for future research. Infact in this project we performed comparison with set value of T and β, but possibly one would like to find a method to include in the algorithm the fixing of these parameters.

One further line of research could be to use this algorithm in the research of wide valley of local minima as proposed in [2], leading to an application in neural networks for which would be necessary generalizing the gradient descent to a Stochastic gradient descent.

Another alternative would be to use a parametrized form of the control instead of the constant u ,maybe time dependent, in order to better estimate the $\nabla F(w)$.

With this project we have shown that the control based gradient descent could prove itself as a valid alternative with further reasearch, maybe even outperforming the algorithm proposed by Chaudhari in [2].

Bibliography

[1] Baldassi, C., Borgs,C., Chayes,J., Ingrosso, A., Lucibello, C., Saglietti, L., and Zecchina, R. (2016). *Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes.* PNAS 113 (48) E7655-E7662, https://doi.org/10.1073/pnas.1608103113

[2] Chaudhari,P., Choromanska,A., Soatto,S., LeCun,Y., Baldassi,C., Borgs,C., Chayes,J., Sagun,L. and Zecchina,R. (2017). *Entropy-SGD: Biasing Gradient Descent Into Wide Valleys.* arXiv:1611.01838

[3] Thijssen, S. and Kappen, H. J. (2015).*Path integral control and state-dependent feedback*. Phys. Rev. E, 91:032104. http://arxiv.org/abs/1406.4026

[4] Fleming, W. H. and Mitter, S. K. (1982). *Optimal control and nonlinear filtering for nondegenerate diffusion processes*. Stochastics: An International Journal of Probability and Stochastic Processes, 8(1):63–77.

[5] Kappen, H. (2005). *Linear theory for control of non-linear stochastic systems.* Physical Review letters, 95:200201

[6] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi Science *Optimization by Simulated Annealing* , New Series, Vol. 220, No. 4598. (May

13, 1983), pp. 671-680

[7] David J. Earl and Michael W. Deem, (2005) *Parallel tempering: Theory, applications, and new perspectives*, Physical Chemistry Chemical Physics,