Experiments with Stochastic Gradient Descent: Condensations of the Real line

Gustavo Lacerda
gusl@cs.ubc.ca

Abstract

It is well-known that training Restricted Boltzmann Machines (RBMs) can be difficult in practice. In the realm of stochastic gradient methods, several tricks have been used to obtain faster convergence. These include gradient averaging (known as momentum), averaging the parameters w^t, and different schedules for decreasing the "learning rate" parameter. In this article, we explore the use of continuous bijective transformations of the parameter space ("condensations"), which effectively amounts to making each parameter's learning rate a function of its current location on the real line. We report on experiments applying condensations to Hinton & Salakhutdinov's (2006) Contrastive Divergence procedure on the MNIST dataset, and show a statistically-significant improvement relative to constant and inverse-log schedules of the learning rate.

1 Introduction

It is well-known that training Restricted Boltzmann Machines (RBMs) can be difficult in practice [1]. The optimization problem consists of finding values of the weights $w_{i,j}$ that minimize an objective function f (often the maximum likelihood).

In the realm of first-order (i.e. gradient-based) methods, several tricks have been used to obtain faster convergence. Firstly, one uses the idea of stochastic gradient descent: instead of computing the gradient on all the data, one does it on a random subset of the data.

Although under common assumptions we are guaranteed to converge to a local minimum of f in the limit, this process can be highly inefficient. One therefore desires to adjust the learning rate appropriately, but this isn't generally easy: if the learning rate is too small, one moves too slowly; but if it is too large, one bounces around too much. Within the framework of stochastic gradient descent, further tricks include averaging the w [3], averaging the gradient increment (known as "momentum") [5], and different schedules for decreasing the "learning rate" parameter.

In this article, we report on experiments with Stochastic Gradient Descent, focusing on bijective continuous transformations of the real line (known as "condensations").

2 Stochastic Gradient Descent

The motivation behind Stochastic Gradient Descent is that when using large samples, computing the gradient becomes expensive, and it becomes cheaper to use mini-batches instead (i.e. the loss is only evaluated on a fraction of the data). If this is done reasonably, the gradient computed at most steps will be a good enough approximation to the true gradient, and large sample results still guarantee almost-sure convergence to a local minimum.

The simplest update procedure is known as the Widrow-Hoff rule [6]. The following update is performed:

$$w^{t+1} := w^t + \eta \nabla_w f(w^t). \tag{1}$$

where the w has been flattened to a column vector, so that $\nabla_w f = (\frac{df}{dw_{1,1}}, ..., \frac{df}{dw_{m,n}})^T$.

η is called the "learning rate", and $\eta \nabla_w f(w^t)$ is known as the "increment" (keep in mind that the derivatives here are stochastic approximations of the true derivative).

To write this component-wise, for each i, j, we perform:

$$w_{i,j}^{t+1} := w_{i,j}^t + \eta \frac{df}{dw_{i,j}}(w_{i,j}^t) \tag{2}$$

We can apply this procedure to the Contrastive Divergence gradient [1] (also known as the "reconstruction error gradient", a more quickly computable approximation to the log-likelihood gradient). With this gradient, our increments push us in the direction of minimizing the reconstruction error.

3 Experiments with the learning rate

In this article, we run experiments by modifying Hinton and Salakhutdinov's (2006) code for training RBMs with Contrastive Divergence on the MNIST handwritten digits dataset (10 000 images, 728 pixels each) [1]. This code uses a Widrow-Hoff stochastic gradient procedure, which divides the data into 100 mini-batches, and its gradient step includes a momentum term. On all our experiments, unless indicated otherwise, the momentum term was removed, and 10 hidden nodes were used.

As we can see in Fig. 1, the effect of momentum seems to be significant but not huge.

3.1 Experiment 1 - Selecting a schedule for the learning rate

Each descent was run for 100 epochs (a.k.a iterations, or time steps). This experiment has 3 conditions and 5 starting points. We compare the reconstruction error curves obtained by the following schedules of the learning rate. As a baseline learning rate, we used $\eta_0 = 0.1$, following Hinton's default. Each schedule was designed so that at the 10th iteration $\eta = \eta_0$:

- (a) the constant schedule, used in Hinton's code: $\eta(t) = \eta_0$
- (b) the inversely proportional schedule: $\eta(t) = \eta_0 \frac{10}{t}$
- (c) the inverse logarithmic schedule: $\eta(t) = \eta_0 \frac{log(11)}{log(t+1)}$.

By looking at the graphs in Figure 2, we see that the inversely proportional schedule performs badly from all 5 starting points, and that although the inverse logarithmic schedule seems to get a head-start, the constant schedule eventually catches up on every one of these instances. In future experiments, we will compare against both the constant and the inverse logarithmic schedules.

3.2 Experiment 2 - Optimizing η_0

Using the inverse logarithmic schedule, we experimented with different values of η_0.

By looking at graphs in Figure 3, we conclude that the best value of η is about 0.05 (Out of 5 random starting points, $\eta = 0.05$ was arguably the best every time).

4 Condensations of the real line

A natural variation of the Widrow-Hoff idea is to compute the gradient on a transformed version of the parameter space. For example, Kivinen and Warmuth's (1997) Exponentiated Gradient method [4] consists of a continuous transformation of $\mathbf{R}$ (namely the exponential function). However, this

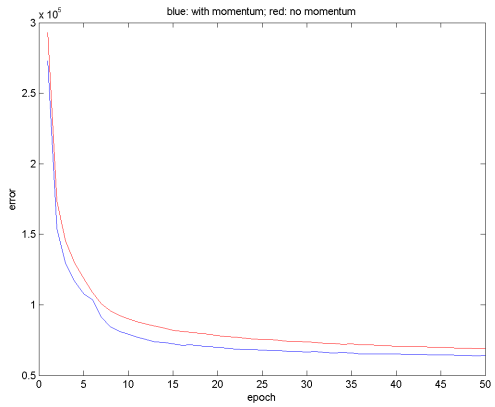

Figure 1: The blue curve shows reconstruction error in Hinton's original code, with momentum. The red curve shows the descent starting from same starting point, but without momentum.

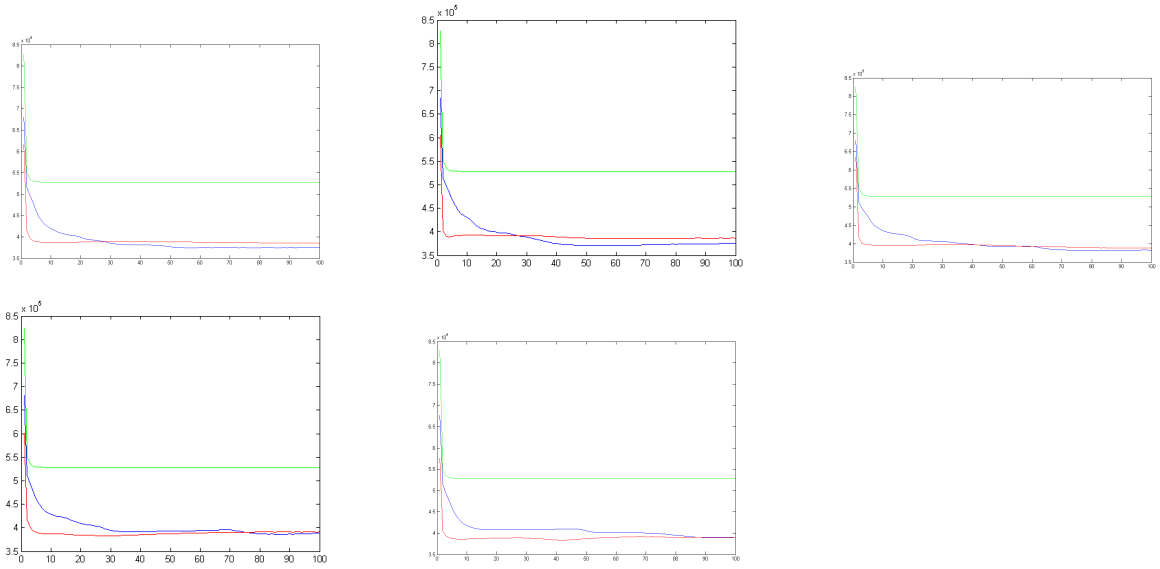

Figure 2: Number of epochs vs reconstruction error. Blue is the constant schedule. Red is the inverse logarithmic schedule. Green is the inversely proportional schedule.

transformation is not bijective, since its range is the positive reals, and therefore the procedure never converges to negative numbers, even if that is where the minimum lies (even if f were convex). To handle this issue, an extension of EG, known as EG$\pm$, is also proposed. In the present article, however, we wish to propose a simpler solution and a more general framework.

Here we consider continuous *bijective* transformations from $\mathbf{R}$ to $\mathbf{R}$. Continuous bijective functions are known as "condensations", and are strictly monotonic. If we apply the Widrow-Hoff idea on condensations of the parameter space, we never lose convergence. We will, among others, use a special case of condensation based on the exponential function, which we call the "oddly-symmetrized Exponential function". Besides the above properties, condensations of the real line also form a group under composition, with the increasing condensations as a subgroup.

4.1 Update Procedure

Let $h : \mathbf{R} \to \mathbf{R}$ be a continuous monotonic bijective function.
Now, w is updated in 4 steps:

$$y^t := h(w^t) \tag{3}$$

$$\nabla_y f(y^t) := \frac{dw}{dy} \nabla_w f(w^t) \tag{4}$$

$$y^{t+1} := y^t + \eta \nabla_y f(y^t) \tag{5}$$

$$w^{t+1} := h^{-1}(y^{t+1}) \tag{6}$$

Eqs. 3 and 6 show the transformation to and from the condensed space. Eq. 4 follows from the Chain Rule. Eq. 5 is the step of adding the increment in the condensed space. For notational simplicity, in the paragraph below, we write w to mean $w_{i,j}$.

From this update procedure, one can see that the effect of such transformations is to make the effective learning rate η' dependent on w. More precisely, we have a different effective learning rate for each component, governed by the same function η', such that the learning rate for w is $\eta'(w)$.

Under the normal gradient, the update term for parameter w is $\eta \frac{df}{dw}(w^t)$, so we say that the learning rate is η. Under the transformation, the update will be $\eta \frac{dw}{dy} \frac{df}{dw}(w^t)$, so effectively the learning rate under the transformed update is $\eta'(w) = \eta \frac{dw}{dy} = \frac{1}{h'(w)} \eta$, i.e. the learning rate is scaled by the inverse of the derivative of h: the flatter the h, the bigger the effective learning rate

4.2 Design principles for condensations

Given the above, we wish to design transformations $h : \mathbf{R} \to \mathbf{R}$ according to the following principles:

- h is bijective, so that every real number is reachable.
- h' is continuous, non-negative everywhere, and not identically zero on any open set (so that the scaling of the learning rate is never infinite).
- h' is evenly symmetric around 0. This ensures that the effective scaling of the learning rate treats positive and negatives numbers equally.
- h is analytically differentiable and invertible.

The first two principles are met by every increasing condensation, whereas the latter two are merely desirable properties.

The following subsection reports experiments with different kinds of condensation.

4.3 Some Condensations

Since we want evenly-symmetric first derivatives, we will design oddly-symmetric functions.

4.3.1 The Symmetrized Power

For any positive-valued power p, we define $h_p(x) = \begin{cases} x^p & \text{if } x \geq 0 \\ -|x|^p & \text{otherwise} \end{cases}$

Thus $h'_p(x) = \begin{cases} px^{p-1} & \text{if } x \geq 0 \\ -p|x|^{p-1} & \text{otherwise} \end{cases}$

This is a condensation since it is monotonically increasing and continuous. See Fig. 4 to visualize this condensation for the case where $p = 2$.

A preliminary experiment (see Fig. 5) suggests that Symmetrized Power is not a very promising condensation, since the Symmetric Power with $p = 1$ (i.e. the identity function) performed better than $p =$ 1/3, 1/2, 2 and 3. For $p > 1$, this may be because the learning rate goes to infinity as w approaches 0 (adding a small positive constant to the first derivative might solve this). For $p < 1$, the implied learning rate encourages the parameters to go away from 0, which is the opposite of what one would desire, by the principles of regularization.

4.3.2 The Symmetrized Exponential

We define $h_{\exp}(x) = \begin{cases} e^x - 1 & \text{if } x \geq 0 \\ -e^{|x|} + 1 & \text{if } x < 0 \end{cases}$

It is easy to check that this is a condensation. See Fig. 6 to visualize its shape.

We derive: $h^{-1}_{\exp}(y) = \begin{cases} log(y+1) & \text{if } x \geq 0 \\ -log(1-y) & \text{otherwise} \end{cases}$

and $h'_{\exp}(x) = e^{|x|}$.

We modify the update procedure with $h_{\exp}$, run on 10 random starting points, and compare against normal gradient under constant and inverse-log learning rate schedules. Fig. 7 shows their performance under 3 starting points. Fig. 8 shows Symmetrized Exponential's relative improvement: Symmetrized Exponential had lower reconstruction error on every instance.

One possible explanation for this success is that transformations with derivatives that increase quickly the farther one gets from 0 act as a kind of "search regularization": the reduced learning rate would have an effect similar to an energy barrier, effectively preventing the parameters from getting too large. And unlike Symmetrized Power, the slope is lower-bounded at positive number (in this case, 1), so the scaling of the learning rate is upper-bounded (in this case, at 1).

4.3.3 The Symmetrized Double Exponential

We define $h_{\exp^2}(x) = h_{\exp}(h_{\exp}(x))$. Since condensations are closed under composition, this is a condensation. Preliminary experiments suggest that this condensation is not promising.

5 Conclusions and Future Work

We have presented the general concept of condensation, and applied it to the Widrow-Hoff rule, yielding a stochastic gradient procedure in which the learning rate is effectively scaled for each parameter $w_{i,j}$ at any given time based its value at that time.

For the task of training RBMs on the MNIST dataset with Contrastive Divergence with 10 hidden variables, the Symmetrized Exponential condensed version of the Widrow-Hoff rule under a "constant schedule" seems to provide a statistically significant improvement, relative to constant and inverse-log learning rate schedules of the normal gradient, when the baseline learning rate $\eta_0 = 0.1$.

Despite these results, one should bear in mind that the benefits of faster convergence might not be worth the cost associated with a 4-step update procedure: although we may need fewer iterations to reach the same level of reconstruction error, each iteration will take longer.

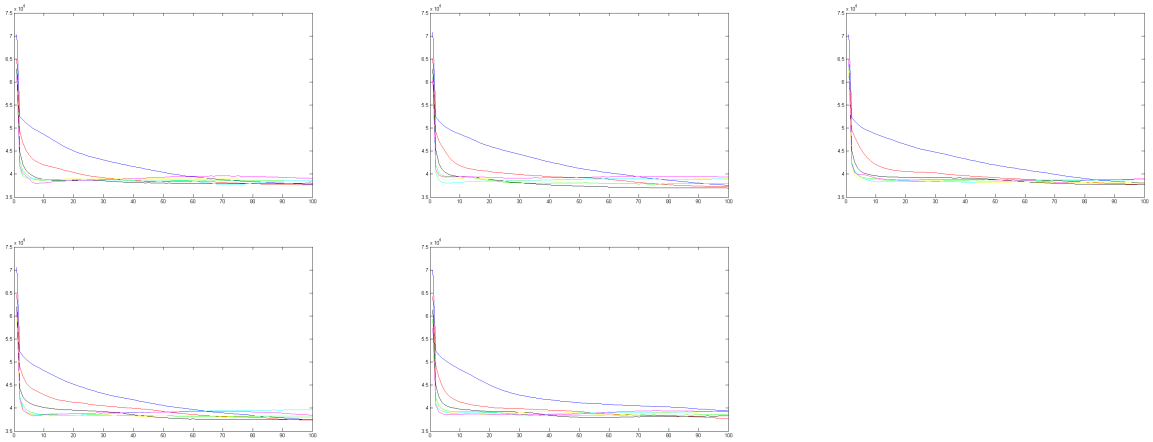

Figure 3: Number of epochs vs reconstruction error. Each color represents a different setting of η_0: blue 0.01, red 0.03, black 0.05, green 0.07, yellow 0.09, cyan 0.11, magenta 0.13. The black curve is arguably the best for all 5 starting points.

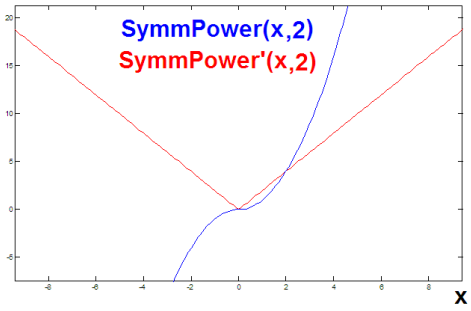

Figure 4: An oddly-symmetric version of a parabola.

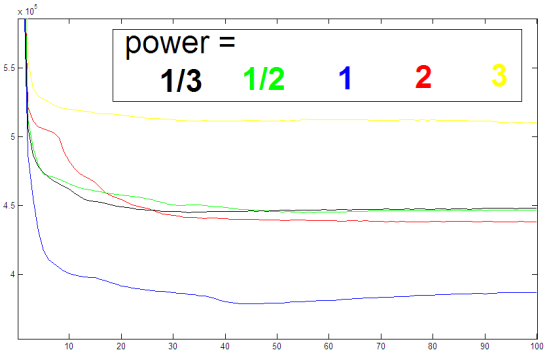

Figure 5: Each color represents a Symmetric Power. "1" corresponds to the usual Widrow-Hoff gradient with a constant learning rate.

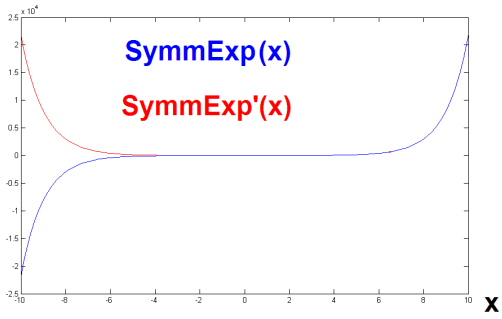

Figure 6: The blue curve shows the symmetrized exponential. The red curve shows its derivative. Note that this function satisfies all our desiderata for a condensation of the reals: h is a continuous bijection of $\mathbf{R} \to \mathbf{R}$ and analytically invertible and differentiable, h' is evenly symmetric, non-negative everywhere.

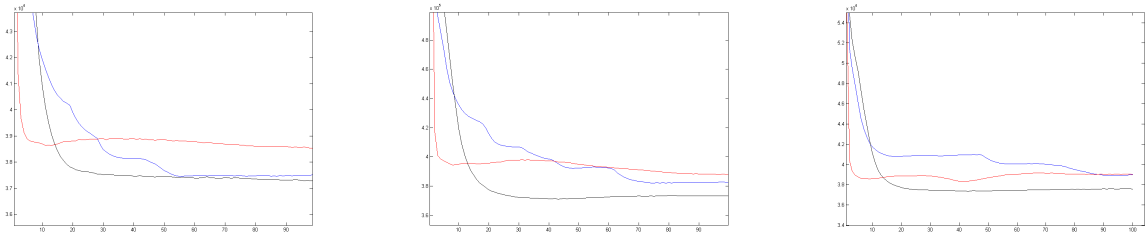

Figure 7: In these 3 instances, we show the behavior of Symmetrized Exponential (black) against normal Widrow-Hoff gradient descent under constant (blue) and inverse-log (red) schedules. $\eta_0 = 0.1$

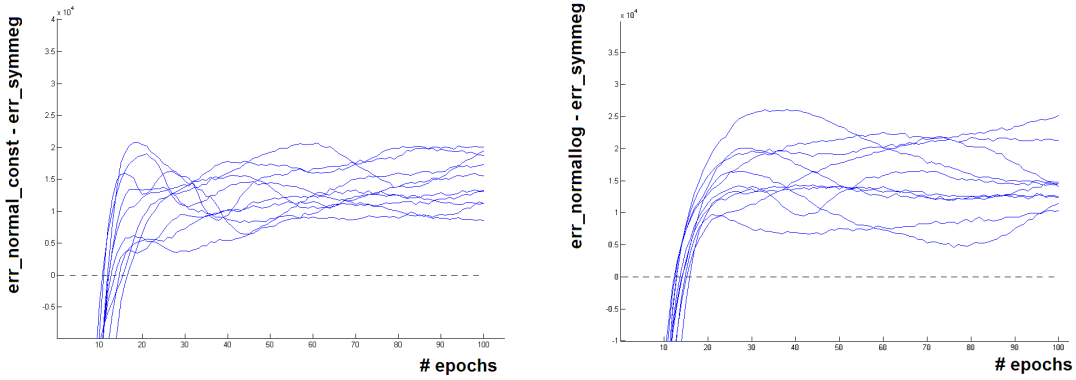

Figure 8: In 10 random starting points, the symmetrized exponential condensation always has lower reconstruction error than normal gradient. These plots show number of epochs vs difference in reconstruction error. The left plot compares it against the constant learning rate schedule, the right plot against the inverse-log schedule.

As for future work, there are several ideas towards the goal of efficient convergence, even while restricting ourselves to first-order methods:

- Do experiments with more hidden nodes. On the MNIST data, it is more typical to use hundreds of hidden nodes [1].
- It may be possible, for some condensations, to optimize the condensed update procedure, by finding a way to work in the condensed space directly (e.g. sample the hidden layer), saving the trouble of having to perform steps 1 and 4.
- Learning rate schedules adapt the learning rate to the epoch. Transformations, such as condensations, adapt it to the current value of the parameter. It would be interesting to combine both.
- There are clearly many parameters to experiment with (learning rate schedules, condensations, momentum, averaging, number of mini-batches), and a more systematic exploration would be desirable. An automatic configurator such as ParamILS [2] would be useful there.
- Adaptively adjust the number of mini-batches. It might be the case that there are places and times in the search when we want a smaller variance in the increment.
- It would be interesting to try to learn when to use which condensations, perhaps by a compositional grammar of condensations, constructing the condensations that are predicted to work best at any given problem and time. As usual, we would need to be careful to prevent overfitting to our dataset and circumstances.

Acknowledgements

Thanks to CPSC550 colleagues, especially Kevin Swersky, for help with the Hinton code; and to George Schaeffer for suggesting the term "condensation".

References

[1] G. E. Hinton, R. R. Salakhutdinov (2006) - Reducing the dimensionality of data with neural networks. *Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006*

[2] F. Hutter, H. H. Hoos, T. Stützle (2007) - Automatic algorithm configuration based on local search. *In Proc. of the Twenty-Second Conference on Artifical Intelligence (AAAI '07)*

[3] A. Juditsky, A. V. Nazin, A. B. Tsybakov, N. Vayatis (2005) - Generalization Error Bounds for Aggregation by Mirror Descent with Averaging. *Advances in Neural Information Processing Systems*

[4] J. Kivinen, M. Warmuth (1997) - Exponentiated Gradient versus Gradient Descent for Linear Predictors, *Information and Computation, Vol. 132, No. 1. (10 January 1997)*

[5] T. K. Leen, G. B. Orr (1994) - Optimal Stochastic Search and Adaptive Momentum, *Advances in Neural Information Processing Systems 6*

[6] B. Widrow, M. E. Hoff (1960) - Adaptive switching circuits *In 1960 IRE WESCON Convention Record*