Adam: Adaptive Moment Estimation

The material here is based on a paper by D. P. Kingma and J. L. Ba, the inventors of the Adam technique.

The error to be minimized

Training data is given as the pairs (x_i, y_i), $i = 1, \ldots, m$, where x_i is a feature vector and y_i is the desired prediction. Let f be the function being used to predict y_i from x_i. It depends on weights, specified by the weights vector W. Let e be the error penalty, which also depends on W. For example, e can be defined as: $e(W, x, y) = (y - f(x, W))^2$. The error penalty for the ith training example is:

$$e_i = e(W, x_i, y_i)$$

The error penalty for the entire training data can be defined as: $E(W) = \frac{1}{m} \sum_{i=1}^{m} e(W, x_i, y_i)$. It is customary to add a regularization term $r(W)$ to make sure that weights do not become too big. For example: $r(W) = |W|^2$. Thus, the error to be minimized is given by:

$$E(W) = \frac{1}{m} \sum_{i=1}^{m} e_i(W, x_i, y_i) + \lambda r(W),$$

where $\lambda \geq 0$ is user defined. To make this error "stochastic", we select at time t a batch of b random examples out of the m given examples, where typically $b \ll m$. The stochastic error to be minimized is:

$$E_t(W) = \frac{1}{b} \sum_{i=1}^{b} e_i(W, x_i, y_i) + \lambda r(W),$$

Stochastic Steepest Descent

The technique of stochastic steepest descent runs many iterations indexed by t. For each t there is a single weights update that uses steepest descent to minimize the stochastic error E_t:

$$W_t = W_{t-1} - \epsilon \nabla E_t(W_{t-1})$$

This can be written as the following three step process that computes W_t from W_{t-1}:

1. $D_t = \nabla E_t(W_{t-1})$

2. $C_t = -\epsilon D_t$

3. $W_t = W_{t-1} + C_t$

where C_t is the change of the weights vector at time t.

Old Momentum

The intuitive idea behind momentum is that the vector C_t should not be changing too rapidly. This can be expressed as $C_t \approx C_{t-1}$. As we see next, an intuitive implementation of this idea creates a mechanism that automatically adjusts the value of ϵ. Here is the momentum idea:

```
0. $C_0 = 0$
1. for $t = 1, 2, \ldots$
    1.1. $D_t = \nabla E_t(W_{t-1})$
    1.2. $C_t = \alpha C_{t-1} - \epsilon D_t$
    1.3. $W_t = W_{t-1} + C_t$
```

where $0 \leq \alpha < 1$. To understand what's going on here consider a single weight w_t from the vector W_t, and the corresponding values c_t from C_t, and d_t from D_t.

$$\begin{aligned}
c_0 &= 0. \\
c_1 &= -\epsilon d_1 \\
c_2 &= \alpha c_1 - \epsilon d_2 = -\epsilon(\alpha d_1 + d_2) \\
c_3 &= \alpha c_2 - \epsilon d_3 = -\epsilon(\alpha^2 d_1 + \alpha d_2 + d_3) \\
c_k &= -\epsilon(\alpha^{k-1} d_1 + \alpha^{k-2} d_2 + \ldots + d_k)
\end{aligned}$$

Observe that if d_t are mostly positive or mostly negative the contributions to c_k accumulate. For the case where they are all identical, $d_t = d$, we have:

$$c_k \approx -\epsilon \frac{1}{1-\alpha} d, \quad w_k \approx w_{k-1} - \epsilon \frac{1}{1-\alpha} d$$

Thus, this behaves as an increase of ϵ by a factor of $1/(1-\alpha)$. If the gradient sign keeps changing (from + to −) then there is no effective increase in ϵ.

Typical values: $\epsilon = 0.1$, $\alpha = 0.9$.

New Momentum technique: ADAM

The idea: similar to old momentum. Increase the value of ϵ if gradient values have same sign, and decrease it if gradient values keeps changing signs. How can this be determined? The square of gradient values always has same sign. Compute Average gradient: call it $\hat{D}$. Compute Average square of gradient: call it $\hat{R}$. The size comparison should be between $\hat{D}$ and $\sqrt{\hat{R}}$.

$\hat{D}$ large $\sqrt{\hat{R}}$ large - accelerate speed

$\hat{D}$ large $\sqrt{\hat{R}}$ small - impossible

$\hat{D}$ small $\sqrt{\hat{R}}$ large - decelerate speed

$\hat{D}$ small $\sqrt{\hat{R}}$ small - accelerate speed

Conclusion: accelerate proportional to $\hat{D}/\sqrt{\hat{R}}$.

Computing running averages

Here we discuss how to compute approximate running averages. These are averages over previously encountered values, averaged so that the most recent values are dominant.

Consider computing the running average of X_t. Denote this running average by $\overline{X}_t$. The idea is that $\overline{X}_t$ can be calculated as a weighted sum of $\overline{X}_{t-1}$ and X_t. The update rule is:

$$\overline{X}_t = \beta \overline{X}_{t-1} + (1-\beta) X_t$$

This is a difference equation that can be easily solved.

$$\begin{aligned}
&\overline{X}_0 = 0 \\
&\overline{X}_1 = \beta \overline{X}_0 + (1-\beta)X_1 = (1-\beta)X_1 \\
&\overline{X}_2 = \beta \overline{X}_1 + (1-\beta)X_2 = \beta(1-\beta)X_1 + (1-\beta)X_2 \\
&\overline{X}_3 = \beta \overline{X}_2 + (1-\beta)X_3 = \beta^2(1-\beta)X_1 + \beta(1-\beta)X_2 + (1-\beta)X_3 \\
&\overline{X}_4 = \beta \overline{X}_3 + (1-\beta)X_4 = \beta^3(1-\beta)X_1 + \beta^2(1-\beta)X_2 + \beta(1-\beta)X_3 + (1-\beta)X_4 \\
&\overline{X}_t = \beta^{t-1}(1-\beta)X_1 + \ldots + \beta^{t-i}(1-\beta)X_i + \ldots
\end{aligned}$$

$$\overline{X}_t = (1-\beta)\sum_{i=1}^{t} \beta^{t-i} X_i$$

If $X_i = X$, a constant, then: $\overline{X}_t = X(1-\beta)\frac{1-\beta^t}{1-\beta} = (1-\beta^t)X$. But we want this value to be X, and this can be achieved if the value is divided by $(1-\beta^t)$. This suggests the following algorithm for running average:

$$\overline{X}_0 = 0, \quad \overline{X}_t = \beta \overline{X}_{t-1} + (1-\beta)X_t, \quad \hat{X}_t = \frac{\overline{X}_t}{1-\beta^t}$$

The ADAM algorithm

```
0. $\overline{D}_0 = 0$, $\overline{R}_0 = 0$, $W_0$ is the initial weights.
1. for $t = 1, 2, \ldots$
    1.1. $D_t = \nabla E_t(W_{t-1})$.
    1.2. $\overline{D}_t = \beta_1 \overline{D}_{t-1} + (1-\beta_1)D_t$.
    1.3. $\overline{R}_t = \beta_2 \overline{R}_{t-1} + (1-\beta_2)D_t^2$. (The vector $D_t^2$ is created by squaring each coordinate of $D_t$.)
    1.4. $\hat{D}_t = \frac{\overline{D}_t}{1-\beta_1^t}$
    1.5. $\hat{R}_t = \frac{\overline{R}_t}{1-\beta_2^t}$
    1.6 $W_t = W_{t-1} - \alpha \frac{\hat{D}_t}{\sqrt{\hat{R}_t}}$ (Both square root and division are applied separately for each coordinate.)
```

To guard against division by a very small number the last step is typically implemented as:

$$W_t = W_{t-1} - \alpha \frac{\hat{D}_t}{\sqrt{\hat{R}_t + \epsilon}}$$

Typical values:

$$\begin{aligned} 0 < \alpha, &\quad \text{e.g., } 0.001 \\ 0 \leq \beta_1 < 1, &\quad \text{e.g., } 0.9 \\ 0 \leq \beta_2 < 1, &\quad \text{e.g., } 0.999 \\ 0 < \epsilon, &\quad \text{e.g., } 10^{-8} \end{aligned}$$